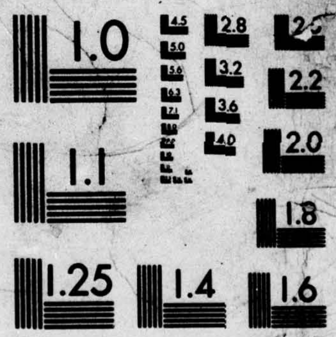
AD-A039 243 RESEARCH IN INFORMATION THEORY(U) LINKABIT CORP SAN DIEGO CALIF J P SCHALKWIJK ET AL. 09 MAY 77 6148-A002 N00123-76-C-0842 1/1

UNCLASSIFIED F/G 9/4 NL


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

Report 6148-A002

RESEARCH IN INFORMATION THEORY

Dr. J. P. Schalkwijk, Eindhoven University of Technology
K. A. Post, Eindhoven University of Technology
A. J. Vinck, Eindhoven University of Technology
LINKABIT Corporation
10453 Roselle Street
San Diego, CA 92121

9 May 1977

Scientific and Technical Report - FINAL
Period 12 April 1976 - 11 April 1977

DISTRIBUTION

Approved for public release, distribution unlimited.

Prepared for

NAVAL REGIONAL PROCUREMENT OFFICE
Long Beach, CA 90822

Delivered to

SUPPLY OFFICER
NAVAL ELECTRONICS LABORATORY CENTER
271 Catalina Blvd.
San Diego, CA 92152

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
6148-A002

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
RESEARCH IN INFORMATION THEORY

5. TYPE OF REPORT & PERIOD COVERED
Scientific & Technical Report - Final
12 April 76 to 11 April77

6. PERFORMING ORG. REPORT NUMBER
6148-A002

7. AUTHOR(s)

J. P. M. Schalkwijk
K. A. Post
A. J. Vinck
Eindhoven University of Technology

8. CONTRACT OR GRANT NUMBER(s)
N00123-76-C-0842

9. PERFORMING ORGANIZATION NAME AND ADDRESS
LINKABIT Corporation
10453 Roselle Street
San Diego, CA 92121

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
XR02105
XR0210501
N438

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Regional Procurement Office
Long Beach, CA 90822

12. REPORT DATE
9 May 77

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Commander DCASD San Diego
Bldg 4, AF Plant 19
4297 Pacific Highway
San Diego, CA 92110

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
--

16. DISTRIBUTION STATEMENT (of this Report)
Distribution A — Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Same

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
None

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper concerns a state space approach to syndrome decoding of binary rate-k/n convolutional codes. State space symmetries of a certain class of codes can be exploited to obtain an exponential reduction of decoder hardware. Aside from these hardware savings, it is felt that the state space formalism developed in this report has some intrinsic value of its own.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

This paper concerns a state space approach to syndrome decoding of binary rate-k/n convolutional codes. State space symmetries of a certain class of codes can be exploited to obtain an exponential reduction of decoder hardware. Aside from these hardware savings, it is felt that the state space formalism developed in this report has some intrinsic value of its own.

TABLE OF CONTENTS

1.0 INTRODUCTION

At the starting point of the present line of research is an optimal coding strategy for the binary symmetric channel with noiseless feedback, see J.P.M. Schalkwijk, "A Class of Simple and Optimal Strategies for Block Coding on the Binary Symmetric Channel with Noiseless Feedback," IEEE Transactions on Information Theory, Vol. IT-17, pp. 283-287, May, 1971. This coding strategy is optimal in that it achieves the minimum possible error probability for a given transmission rate. In addition, it can be shown that this minimal achievable error probability can be realized with a decoder that meets the lower bound on the computational complexity for the particular task at hand. The only drawback of our coding strategy is that it requires as side information at the transmitter a (possible delayed) copy of channel noise. This side information can, as in the above reference, be provided by a noiseless feedback channel. Returning the received information via the feedback channel and comparing it with what was transmitted one obtains a (delayed) copy of the forward noise.

It was soon realized that the requirement of a noiseless feedback channel posed a severe constraint on the use of our coding strategy in many practical situations. Assuming the availability of a duplex channel consisting of a BSC in each direction, one can obtain a copy of the forward noise at the active station by returning a linearly scrambled version of the received data, see IEEE Transactions on Communications, Vol. COM-22, pp. 1369-1374, September, 1974. In the Appendix, it is explained how the linear forward scrambler of the above reference was replaced by a more effective feedback scrambler. We thus obtain an extremely efficient strategy for error control on duplex channels.

The duplex strategy has an even more important aspect aside from efficient error control. Assume a multiple dialogue system (MDS) consisting of a central computer with, for example, 25 satellite computers. Operating the communication from the central facility towards a satellite in the traditional one-way mode would require 25 decoders, one at each satellite computer. Using our duplex strategy for communicating from the center towards the satellites requires one single decoder at the central facility. This decoder also handles the information flow from a satellite computer to the central facility. Hence, using our duplex strategy in the above computer communication network one saves 25 decoders, i.e., as many as there are satellite computers.

An important part of the duplex strategy, see IEEE Transactions on Communications, Vol. COM-22, pp. 1369-1374, September, 1974, is the circuit that forms an estimate at the active station of the forward noise. Aside from being a crucial part of the duplex scheme this circuit is important in that it can also be used for data compression purposes and as the core part of a syndrome decoder for binary rate 1/2 convolutional codes, as explained in the Appendix. We were struck by the imbalance in hardware complexity between the feedback strategy as implemented at a satellite station, and the hardware complexity of the syndrome decoder at the central terminal. Hence, our main research effort as described in Section 2.0 has been aimed at reducing the complexity of syndrome decoders for binary rate k/n convolutional codes. Aside from the results on reduced decoder hardware, Section 2.0 is important in that it introduces a state space formalism that appears to be a valuable tool in the theory of convolutional codes.

Section 3.0 concerns syndrome decoding with soft decisions. In practice, when using Viterbi decoding on satellite channels roughly 2 dB in signal-to-noise ratio can be gained by going from 2 to 8 level quantization at the receiver. The aim of the research effort described in Section 3.0 was to investigate how much of the hardware savings obtained in the Appendix for syndrome decoding can be salvaged in the case of soft decisions. The results are rather discouraging and it it dubious if for 8 level quantization at the receiver, one should still prefer syndrome decoding over regular Viterbi decoding. We are presently investigating the possibilities of syndrome decoding over FG(3), but it is still too early to draw any conclusions.

Section 4.0, finally, describes a method to exactly evaluate Viterbi's union bound on the first event error probablity and on the bit error probablity for maximum likelihood decoding of convolutional codes. These results present a further improvement on an earlier bounding technique described by L. v.d. Meeberg, see IEEE Transactions on Information Theory, Vol. IT-20, pp. 389-391, May, 1974. The results of Section 4.0 will appear in the IEEE Transactions on Information Theory, March, 1977.

2.0 COMPLEXITY OF DECODERS FOR CONVOLUTIONAL CODES

SYNDROME DECODING OF BINARY RATE-k/n CONVOLUTIONAL CODES

J.P.M. Schalkwijk, senior member, A.J. Vinck, member,
and K.A. Post, member.

March 1, 1977.

J.P.M. Schalkwijk, and A.J. Vinck are with the Department of Electrical Engineering, and K.A. Post is with the Department of Mathematics, Eindhoven University of Technology, Eindhoven, The Netherlands.

I. INTRODUCTION

This paper concerns a state space approach to syndrome decoding of binary rate-k/n convolutional codes. It extends and generalizes earlier work [1, 2, 3] on syndrome decoding of binary rate-½ convolutional codes. In Sections II, and III we develop a concise mathematical formulation of the problem. Section IV introduces a special class of binary rate-(n-1)/n convolutional codes. It is shown that the state space symmetries of this class of codes allow for an exponential reduction of decoder hardware. Section V extends the results of the previous section to rate-k/n codes. Table I lists the free distance of some short constraint length codes that exhibit the required symmetries.

Fig. 1 shows a conventional [4] binary rate-2/3 convolutional encoder with 2 memory elements. The input to this encoder are

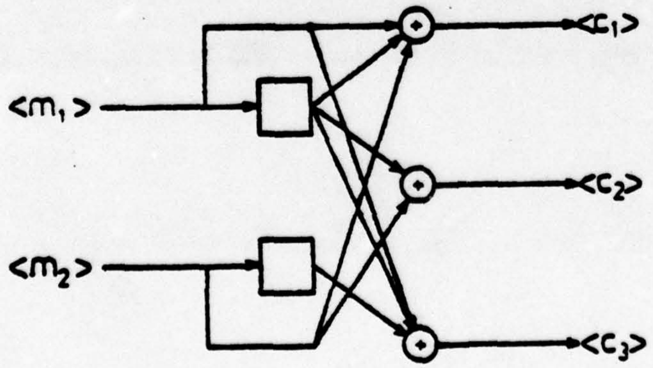


Fig. 1. A rate-2/3 convolutional encoder.

two binary message sequences

$$<m_i> = \ldots, m_{i,-1}, m_{i0}, m_{i1}, \ldots \quad ; i = 1,2 .$$

The outputs are three binary codeword sequences $<c_1>$, $<c_2>$, and $<c_3>$ (hence the rate is 2/3). The elements of the three output sequences $<c_1>$, $<c_2>$, and $<c_3>$ are, respectively,

$$\begin{aligned} c_{1,t} &= m_{1,t} \oplus m_{1,t-1} \oplus m_{2,t} \\ c_{2,t} &= m_{1,t-1} \oplus m_{2,t} \\ c_{3,t} &= m_{1,t} \oplus m_{1,t-1} \oplus m_{2,t-1} \quad , \end{aligned}$$

where $\oplus$ denotes modulo 2 addition.

With the input and output sequences, we associate sequences in the delay operator X:

$$m_i(X) = \ldots + m_{i,-1}X^{-1} + m_{i,0} + m_{i1}X + m_{i2}X^2 + \ldots \quad ; i = 1,2$$

$$c_j(X) = \ldots + c_{j,-1}X^{-1} + c_{j0} + c_{j1}X + c_{j2}X^2 + \ldots \quad ; j = 1,2,3,.$$

For notational convenience we shall generally suppress the parenthetical X in our subsequent references to sequences; thus m_i means $m_i(X)$, $c_j = c_j(X)$, and so forth, where the fact that a letter represents a sequence (transform) should be clear from the context. Now the input/output relationships are expressed concisely as

$$\underline{c} = \underline{m}G \quad , \tag{1}$$

where $\underline{m} = (m_1,m_2)$, $\underline{c} = (c_1,c_2,c_3)$, and the generator matrix $G = [g_{ij}(X)]$ is

$$G = \begin{bmatrix} 1+X & X & 1+X \\ 1 & 1 & X \end{bmatrix} ,$$

and formal power series multiplication with coefficient operations modulo 2 is applied. In general, let there be k inputs and n outputs. If we define the constraint length for the i-th input as

$$\nu_i = \max_{1 \leq j \leq n} \; [\deg g_{ij}(X)] ,$$

then the overall constraint length

$$\nu = \sum_{i=1}^{k} \nu_i ,$$

(ν=2 for the encoder of Fig. 1), equals the number of memory elements for what Forney [4] calls the obvious realization of the encoder.

The dual, $C^{\perp}$, code [5] to a convolutional code C is the linear space generated by the set of all n-tuples of finite (for infinite sequences the inner product may not be defined) sequences $\underline{d}(X)$ such that the inner product $(\underline{c},\underline{d}) \triangleq \underline{c}.\underline{d}^T$ (where T means transpose) is zero for all $\underline{c}$ in C. The dual code of a rate-k/n convolutional code, generated by an encoder G, is a rate-(n-k)/n code that can be generated by a suitable encoder H, such that $GH^T = 0$. The matrix H^T can be obtained from the inverse of the B matrix in an invariant factor decomposition [4, 5], $G = A\Gamma B$, of the encoder matrix G by taking the last n-k columns of B^{-1}. The n-input, (n-k)-output linear sequential circuit whose transfer function matrix is H^T is called a syndrome former, and has the property that $\underline{c}H^T = 0$ if and only if $\underline{c} \epsilon C$.

For the encoder, G, of Fig. 1 we have an invariant factor decomposition

$$\begin{bmatrix} 1+X & X & 1+X \\ 1 & 1 & X \end{bmatrix} = \begin{bmatrix} X & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1+X+X^2 \\ 0 & 0 & 1 \end{bmatrix}$$

Hence,

$$B = \begin{bmatrix} 1 & 1 & X \\ 1 & 0 & 1+X+X^2 \\ 0 & 0 & 1 \end{bmatrix}, \text{ so } B^{-1} = \begin{bmatrix} 0 & 1 & 1+X+X^2 \\ 1 & 1 & 1+X^2 \\ 0 & 0 & 1 \end{bmatrix} .$$

The H^T matrix is now given by the last column of the B^{-1} matrix, i.e.

$$H^T = \begin{bmatrix} 1+X+X^2 \\ 1+X^2 \\ 1 \end{bmatrix}$$

Fig. 2 gives the obvious realization of the syndrome former. Two comments are in order. First, note that for rate-(n-1)/n codes the syndrome former has n inputs but a single output, compare Fig. 2.

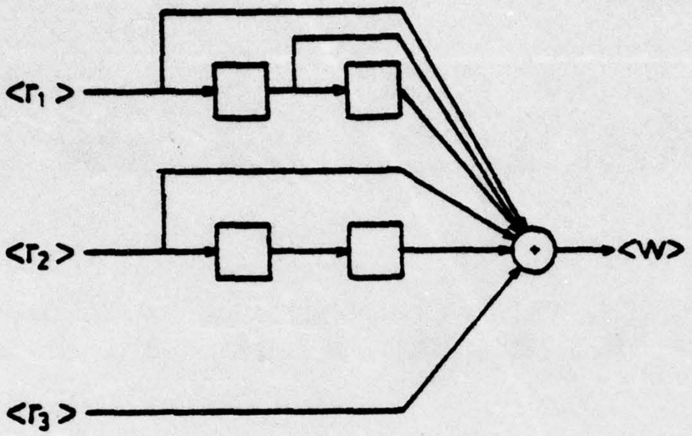


Fig. 2. A syndrome former for a rate-2/3 convolutional code.

This single output is the reason that in Sections II, III, and IV we first concentrate on rate-(n-1)/n codes. Second, in Table II of Section V we list codes in terms of their syndrome formers. The invariant factor theorem can now be used on the matrix H, i.e. $H = C\Gamma D$, to find from the D^{-1} matrix a suitable encoder G. This encoder is conventional (i.e. it has no feedback), but it is not necessarily minimal [4], i.e. the obvious realization does not necessarily have the smallest possible number of memory elements.

Let $\underline{e}(X)$ be the error vector sequence, and let $\underline{r} = \underline{c} + \underline{e}$ be the received data vector sequence. We then define the syndrome vector sequence $\underline{\omega}(X)$ as

$$\begin{aligned} \underline{\omega} &\triangleq \underline{r}H^T \\ &= (\underline{c} + \underline{e})H^T = \underline{e}H^T . \end{aligned}$$

The task of the codeword estimator [4] is now to find an error vector sequence estimate $\underline{\hat{e}}(X)$ of minimum Hamming weight that can be a possible cause of the syndrome vector sequence $\underline{\omega}(X)$. The codeword vector sequence estimate $\underline{\hat{c}}(X)$ is then given by

$$\underline{\hat{c}} = \underline{r} + \underline{\hat{e}} .$$

Using the codeword vector sequence estimate $\underline{\hat{c}}(X)$, the inverse encoder G^{-1} now forms an estimate $\hat{m}(X)$ of the message vector sequence $\underline{m}(X)$, i.e.

$$\underline{\hat{m}} = \underline{\hat{c}}G^{-1} ,$$

where G^{-1} is a right inverse of G, i.e. $GG^{-1} = I$. This inverse encoder, G^{-1}, can also (i.e. like the syndrome former) be obtained from the invariant factor decomposition $G = A\Gamma B$ of the encoder G. For the encoder G of Fig. 1 we have

$$G^{-1} = B^{-1}\Gamma^{-1}A^{-1} = \begin{bmatrix} 0 & 1 & 1+X+X^2 \\ 1 & 1 & 1+X^2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & X \end{bmatrix} = \begin{bmatrix} 1 & X \\ 1 & 1+X \\ 0 & 0 \end{bmatrix}$$

Fig. 3 gives the obvious realization of the inverse encoder G^{-1}.

Note that both G, and G^{-1} represent one-to-one (and in fact linear) maps that can be realized with simple circuitry, compare Figs. 1, and 3. The codeword estimator determines both the complexity and the performance of the system. Section II deals with the state

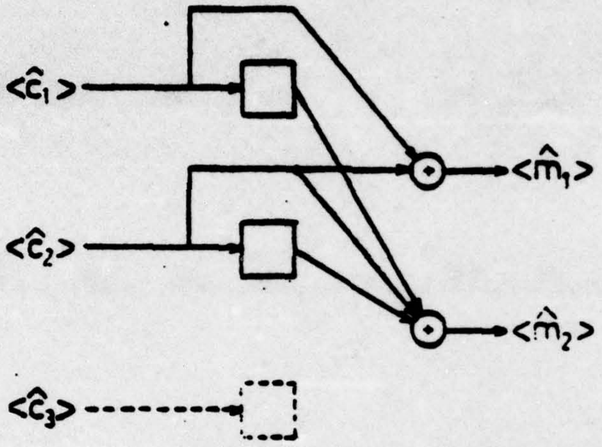


Fig. 3. An inverse encoder for the rate-2/3 convolutional code of Fig. 1.

space of the syndrome former of a binary rate-(n-1)/n convolutional code. Section III gives a description of the codeword estimator in terms of the state space framework developed in Section II. As it turns out certain symmetries in the syndrome former state space can be exploited to greatly reduce the complexity of the codeword estimator. This line is persued in the remainder of the paper.

Before embarking on our state space approach (which is the core of this paper) towards the codeword estimator one final comment is in order. The estimate $\underline{\hat{m}}(x)$ of the message vector sequence $\underline{m}(X)$ can also be written as

$$\underline{\hat{m}} = \underline{\hat{c}}G^{-1} = \underline{r}G^{-1} + \underline{\hat{e}}G^{-1} .$$

The first term $\underline{r}G^{-1}$ on the RHS of above eqn. can be easily obtained from the received data vector sequence $\underline{r}(X)$ using the simple circuitry of Fig. 3. As in refs. [1, 2, 3], it turns out that the overall decoder requires less hardware if we let the estimator determine the second term, $\underline{\hat{e}}G^{-1}$, directly. Hence, we define the message (as opposed to the codeword) vector sequence correction, $\underline{\hat{e}}_m(X)$, as

$$\underline{\hat{e}}_m \triangleq \underline{\hat{e}}G^{-1} . \tag{2}$$

II. STATE SPACE

For a state space analysis it is convenient to represent the syndrome former of a rate-(n-1)/n code by an n-tuple (A,B,C,...,D) of binary polynomials, see Fig. 4. The n-tuple (A,B,C,...,D) is obtained from the matrix $H = [h_1(X), h_2(X), \ldots, h_n(X)]$ of Section I by putting $a_i = h_{1i}$, $b_i = h_{2i}$, $c_i = h_{3i}, \ldots, d_i = h_{ni}$, $i=0,1,2,\ldots,h$, where

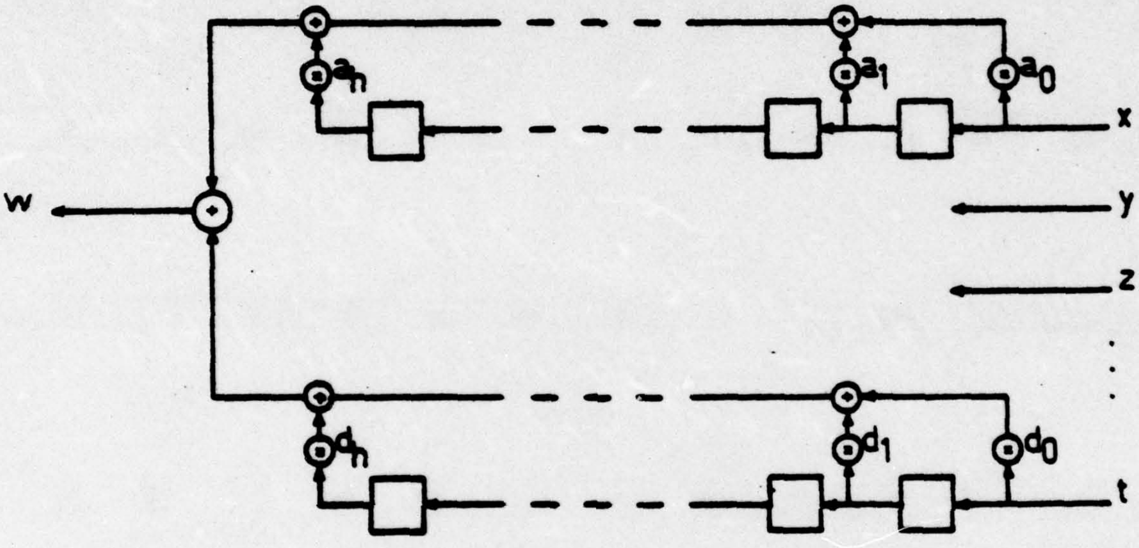


Fig. 4. The syndrome former for a rate-(n-1)/n convolutional code.

$$h = \max_{1 \le j \le n} \deg h_j(X) .$$

Obviously, one single noise vector in the sequence $\ldots, [e_{1,-1}, e_{2,-1}, \ldots, e_{n,-1}]^T$, $[e_{10}, e_{20}, \ldots, e_{n0}]^T$, $[e_{11}, e_{21}, \ldots, e_{n1}]^T, \ldots$ can at most influence h+1 successive syndrome digits. We define the "physical state" of the system to be the nh-dimensional binary vector representing the contents of all shift register stages in Fig. 4. Every noise vector that enters the system causes a transition of its physical state and gives rise to a binary syndrome digit. The phenomenom occurs that two different initial physical states are syndrome-indistinguishable, i.e. that under every noise vector sequence $[e_{10},e_{20},\dots,e_{n0}]^T,[e_{11},e_{21},\dots,e_{n1}]^T,\dots$ their syndrome sequences are identical. It is left to the reader [3,4] to prove that this natural concept of syndrome-indistinguishability is exactly the same as the following equivalence relation: Two physical states are called equivalent if their difference has a sequence of syndrome digits identically zero under a sequence of noise vectors identically zero. In fact, we may restrict ourselves in this definition to sequences of zero-noise vectors of length h, since all following zero-noise vectors simply must yield zero-syndrome digits.

The equivalence classes of the above equivalence relation will be called "abstract states", of briefly "states" of the system. There are several equivalent state descriptions. In ref. [3] Schalkwijk and Vinck use the contents of the bottom register D of the syndrome former, Fig. 4, as a description of the state. Forney [4] uses the zero-noise syndrome sequence to represent the state. In the present paper we opt for this latter description.

We are now ready to introduce some convenient notation: States (given by their zero-noise syndrome sequence) are denoted by lower case greek letters with a subscript, e.g.

$\sigma_1 \triangleq [s_1,s_2,s_3,\dots,s_{h-2},s_{h-1},s_h]$, and its left shifts

$\sigma_2 \triangleq [s_2,s_3,s_4,\dots,s_{h-1},s_h,0]$,

$\sigma_3 \triangleq [s_3,s_4,s_5,\dots,s_h,0,0]$, and so on.

Occasionally, i.e. if sufficiently many terminating components $s_h, s_{h-1}, \ldots$ vanish, we also write the right shifts, e.g.

$$\sigma_0 \triangleq [0, s_1, s_2, \ldots, s_{h-3}, s_{h-2}, s_{h-1}] \text{ if } s_h = 0,$$
$$\sigma_{-1} \triangleq [0, 0, s_1, \ldots, s_{h-4}, s_{h-3}, s_{h-2}] \text{ if } s_h = s_{h-1} = 0.$$

Finally, we introduce the symbols α_1, β_1, $\gamma_1, \ldots$, δ_1 to denote the generator states of the system, i.e.

$$\alpha_1 \triangleq [a_1, a_2, \ldots, a_h],$$
$$\beta_1 \triangleq [b_1, b_2, \ldots, b_h],$$
$$\gamma_1 \triangleq [c_1, c_2, \ldots, c_h],$$
$$\vdots$$
$$\delta_1 \triangleq [d_1, d_2, \ldots, d_h].$$

Without loss of generality we assume $a_h = 1$. This assumption is justified by the definition of h and implies that the state space has dimension h.

The output of the syndrome former, see Fig. 4, at time t and the state at time t+1 are completely determined by the state σ_1 and the input $[e_{1t}, e_{2t}, e_{3t}, \ldots, e_{nt}]^T$ at time t, t=...,-1,0,+1,... .
As the syndrome former is supposed to be time invariant there is no purpose in retaining the subscript t in the state space analysis. Thus, we denote the syndrome former input by $[x, y, z, \ldots, t]^T$. The corresponding state transition and the output ω are given by

$$\begin{tikzcd} & {[x,y,z,\ldots,t]^T} \arrow[d] \\ \sigma_1 \arrow[r, mapsto] \arrow[dr] & \sigma_2 + x\alpha_1 + y\beta_1 + z\gamma_1 + \ldots + t\delta_1 \\ & \omega = s_1 + xa_0 + yb_0 + zc_0 + \ldots td_0 \, . \end{tikzcd} \qquad (3)$$

Fig. 5 gives the state diagram of the syndrome former of Fig. 2. Solid lines correspond to a syndrome digit ω=0 and dashed lines to a

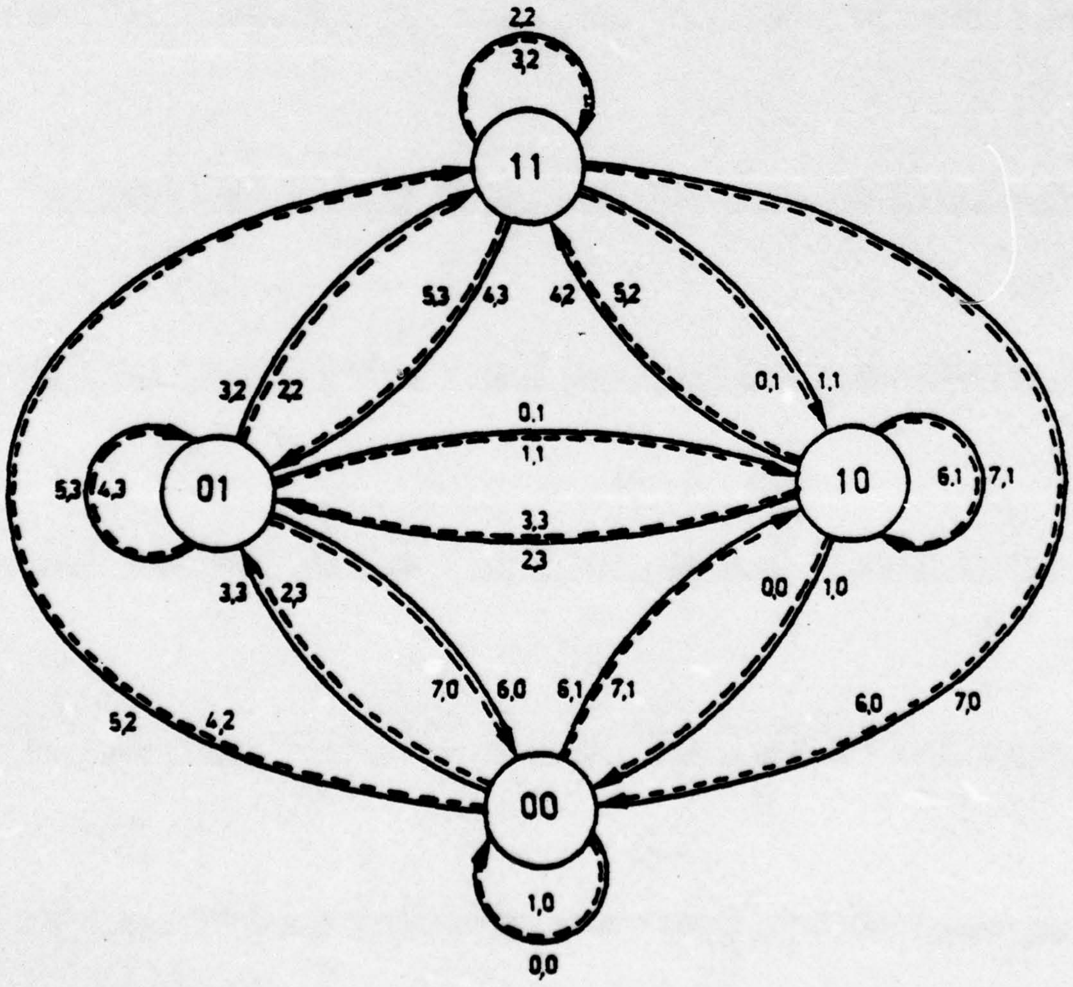


Fig. 5. State diagram of syndrome former.

syndrome digit ω=1. Indicated along the edges are the numerical values of $[\hat{e}_1,\hat{e}_2,\ldots,\hat{e}_n]_2$, $[\hat{e}_{m1},\hat{e}_{m2},\ldots,\hat{e}_{mk}]_2$ intepreted as binary numbers, where the latter vector represents the generic term $[\hat{e}_{m1t},\hat{e}_{m2t},\ldots,\hat{e}_{mkt}]$, t=...,-1,0,+1,... , of the message vector sequence correction $\underline{\hat{e}}_m(X)$ of (2), with the redundant subscript t removed.

The fact that the message vector correction is solely determined by the next state [3], see Fig. 5, follows from Forney [5]. According to Forney the syndrome former state uniquely determines the encoder state and vise versa if G and H^T are connected by a noiseless (dashed in Fig. 6) channel. The vector sequence $\underline{\hat{e}}^t_m(X)$ in Fig. 6 is such that

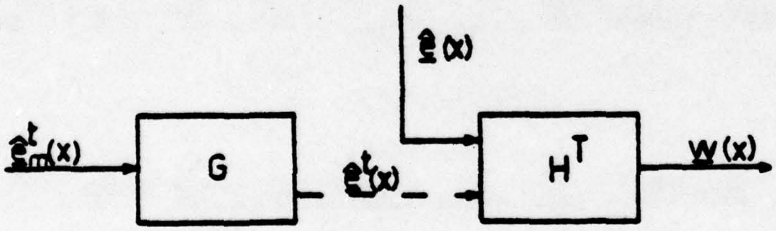


Fig. 6. Encoder and syndrome former connected back to back.

$\underline{\hat{e}}^t = \underline{\hat{e}}^t_m G$ stears the syndrome former H^T to the same state at time t, t=...,-1,0,+1,... , as does $\underline{\hat{e}}(X)$. As we can equate the encoder state to its recent inputs, $[\hat{e}^t_{m1t},\hat{e}^t_{m2t},\ldots,\hat{e}^t_{mkt}]$ is uniquely determined by the state of the syndrome former at time t+1, t=...,-1,0,+1,... . But, as G^{-1} is an instantaneous right inverse of G we have $[\hat{e}_{m1t},\ldots,\hat{e}_{mkt}] = [\hat{e}^t_{m1t},\ldots,\hat{e}^t_{mkt}]$, t=...,-1,0,+1,... .

Now consider the linear subspace spanned by the generators α_1, β_1, $\gamma_1,\ldots,\delta_1$. If this subspace has dimension q then according to (3) each state σ_1 has exactly 2^q state transition images. Again by (3), these images from a coset of the linear subspace $L[\alpha_1,\beta_1,\gamma_1,\ldots,\delta_1]$. This coset will be called the "*sink-tuple*" *of* σ_1.

The linear subspace $L[\alpha_1,\beta_1,\gamma_1,\ldots,\delta_1]$ is identical to the linear subspace $L[\alpha_1,\beta_1+b_h\alpha_1,\gamma_1+c_h\alpha_1,\ldots,\delta_1+d_h\alpha_1]$. However, as $a_h=1$, the vectors $\beta_1+b_h\alpha_1$, $\gamma_1+c_h\alpha_1,\ldots,\delta_1+d_h\alpha_1$ have a rightmost coordinate equal to 0. Thus, these vectors have a right shift. Furthermore,

$$\text{rank}\begin{bmatrix} a_1 & ,\ldots, & a_{h-1} & ,1 \\ b_1+b_ha_1, & \ldots & ,b_{h-1}+b_ha_{h-1}, & 0 \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ d_1+d_ha_1, & \ldots & ,d_{h-1}+d_ha_{h-1}, & 0 \end{bmatrix} = \text{rank}\begin{bmatrix} 1, & 0 & ,\ldots, & 0 \\ 0, & b_1+b_ha_1, & \ldots & ,b_{h-1}+b_ha_{h-1} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ 0, & d_1+d_ha_1, & \ldots & ,d_{h-1}+d_ha_{h-1} \end{bmatrix}$$

Define,

$$\epsilon_1 \triangleq [1,0,0,\ldots,0],$$

a row vector of length h. Then

$$\dim L[\alpha_1,\beta_1,\gamma_1,\ldots,\delta_1] = \dim L[\epsilon_1,(\beta+b_h\alpha)_0,(\gamma+c_h\alpha)_0,\ldots,(\delta+d_h\alpha)_0] .$$

Each state has at least one primage. If $\tau_1 = [s_1,s_2,\ldots,s_{h-1},0]$, then

$\tau_0 = [0,s_1,\dots,s_{h-2},s_{h-1}]$ is a preimage under $[x,y,z,\dots,t] = [0,0,0,\dots,0]$. If $\tau_1 = [s_1,s_2,\dots,s_{h-1},1]$, then$(\tau+\alpha)_0$ is a preimage under $[x,y,z,\dots,t] = [1,0,0,\dots,0]$. But, if a state τ_1 has a preimage then it has at least 2^q preimages, i.e. all the states in the coset of $L[\varepsilon_1,(\beta+b_h\alpha)_0, (\gamma+c_h\alpha)_0,\dots,(\delta+d_h\alpha)_0]$ that contains the particular preimage. We now have the following results. Each state σ_1 has exactly 2^q images, i.e. the sink-tuple of σ_1. On the other hand, each state τ_1 has at least 2^q preimages, i.e. the above mentioned coset of $L[\varepsilon_1,(\beta+b_h\alpha)_0,(\gamma+c_h\alpha)_0, \dots,(\delta+d_h\alpha)_0]$. Hence, we conclude that τ_1 has exactly 2^q preimages that constitute the "source-tuple" of τ_1. It is easily verified that each element σ_1 of a source tuple has the same sink-tuple.

It is this source/sink-tuple description of the state space that will play an important role in the remainder of the paper. Hence, to make things more concrete, we give a specific example for the syndrome former of Fig. 7.

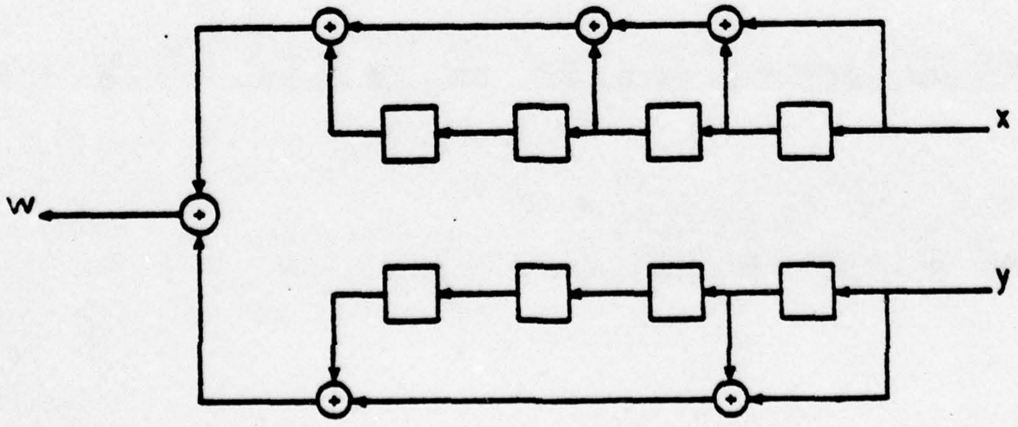


Fig. 7. The syndrome former for a rate-½ convolutional code.

We have

$$\alpha_1 = [1\ 1\ 0\ 1]_2 = 13 \qquad \varepsilon_1 = [1\ 0\ 0\ 0]_2 = 8$$

$$\beta_1 = [1\ 0\ 0\ 1]_2 = 9 \qquad (\alpha+\beta)_0 = [0\ 0\ 1\ 0]_2 = 2$$

Source-tuples		Sink-tuples
{0, 2, 8,10}	$\xrightarrow{I}$	{0, 4, 9,13}
{1, 3, 9,11}	$\xrightarrow{II}$	{2, 6,11,15}
{4, 6,12,14}	$\xrightarrow{III}$	{1, 5, 8,12}
{5, 7,13,15}	$\xrightarrow{IV}$	{3, 7,10,14}

Fig. 8 shows a partition of the state space in source/sink-tuples.

		source-tuples			
		I	II	IV	III
sink-tuples	I	0	9	13	4
	II	2	11	15	6
	IV	10	3	7	14
	III	8	1	5	12

Fig. 8. State space partition in source/sink-tuples.

Anticipating on Section IV the states in Fig. 8 have been geometrically arranged in such a way that also the metric equivalence classes {0}, {4}, {8}, {12}, {9,13}, {6,14}, {1,5}, {2,10}, and {3,7,11,15} are easily distinguishable. Two states that are in the same metric equivalence class have the same metric value [6], irrespective of the received data vector sequence $\underline{r}(X)$.

III. ALGORITHM

Given the syndrome sequence $\omega(X)$ of a rate-$(n-1)/n$ code, compare Fig. 4, the estimator is to determine the state sequence that corresponds to a noise vector sequence estimate $\hat{\underline{e}}(X)$ of minimum Hamming weight that can be a possible cause of $\omega(X)$. According to Section II this state sequence can be stored in terms of an equivalent message vector sequence correction $\hat{e}_m(X)$. As the estimation algorithm to be described in this section is similar to Viterbi's [6], we can be very brief. To find the required state sequence Viterbi introduces a "metric function". A metric function is defined as a nonnegative integer-valued function on the states. With every state transition we now associate the Hamming weight W_H of its noise vector $[x,y,z,\dots,t]^T$.

PROBLEM: Given a metric function f and a syndrome digit ω, find a metric function g which is statewise minimal, and for every state is consistent with at least one of the values of f on its preimages under syndrome digit ω, increased by the weight of its corresponding state transition.

The solution to this problem expresses g in terms of f and ω, and can be formulated in terms of the source/sink-tuples of Section II. In fact, the values of g on a sink-tuple T_i are completely determined by the values of f on the corresponding source-tuple S_i, and by the syndrome digit ω. The equations that express g in terms of f and ω are called "metric equations". They have the form

$$g(\tau_1) = \min \{ f(\sigma_1) + W_H([x,y,z,\dots,t]^T) \mid \sigma_1 \xmapsto[\searrow\ \omega]{[x,y,z,\dots,t]^T\ \downarrow} \tau_1 \} \qquad (4)$$

p. 23 blank

The particular preimage σ_1 in (4) that realizes the *minimum* is called the "survivor". When there are more preimages for which the minimum in (4) is achieved, one could flip a multi-coin to determine the survivor. However, we will shortly discover that a judicious choice of the survivor among the candidate preimages offers the possibility of significant savings in decoder hardware. The construction of (4) can be repeated, i.e. starting with a metric function f_0, given a syndrome sequence $\omega_1,\omega_2,\omega_3,\dots$ one can form a sequence of metric functions $f_1,f_2,f_3,\dots$ iteratively by means of the metric equations:

$$f_0 \xmapsto{\omega_1} f_1 \xmapsto{\omega_2} f_2 \xmapsto{\omega_3} f_3 \longmapsto \dots \ .$$

The metric function f_s, *whose value* $f_s(\sigma_1)$ at an arbitrary state σ_1, equals the Hamming weight of the lightest path from the zero-state to σ_1 under an all zero syndrome sequence, $\omega_1,\omega_2,\omega_3,\dots=0,0,0,\dots$, is called the "stable metric function". It has the property

$$f_s \xmapsto{\omega=0} f_s \ .$$

In order to make things more concrete we now give a specific example. Fig. 9 represents the t-th section, $t=\dots,-1,0,+1,\dots$, of the trellis diagram [6] corresponding to the state diagram of Fig. 5. From Fig. 9 we find for the metric equations:

$$g(0)=\bar{\omega}\ \min[f(0),f(1)+2,f(2)+1,f(3)+3]+\omega\ \min[f(0)+1,f(1)+3,f(2),f(3)+2] \qquad (5a)$$

$$g(1)=\bar{\omega}\ \min[f(0)+2,f(1)+2,f(2)+1,f(3)+1]+\omega\ \min[f(0)+1,f(1)+1,f(2)+2,f(3)+2] \qquad (5b)$$

$$g(2)=\bar{\omega}\ \min[f(0)+2,f(1),f(2)+3,f(3)+1]+\omega\ \min[f(0)+3,f(1)+1,f(2)+2,f(3)] \qquad (5c)$$

$$g(3)=\bar{\omega}\ \min[f(0)+2,f(1)+2,f(2)+1,f(3)+1]+\omega\ \min[f(0)+1,f(1)+1,f(2)+2,f(3)+2]\ , \qquad (5d)$$

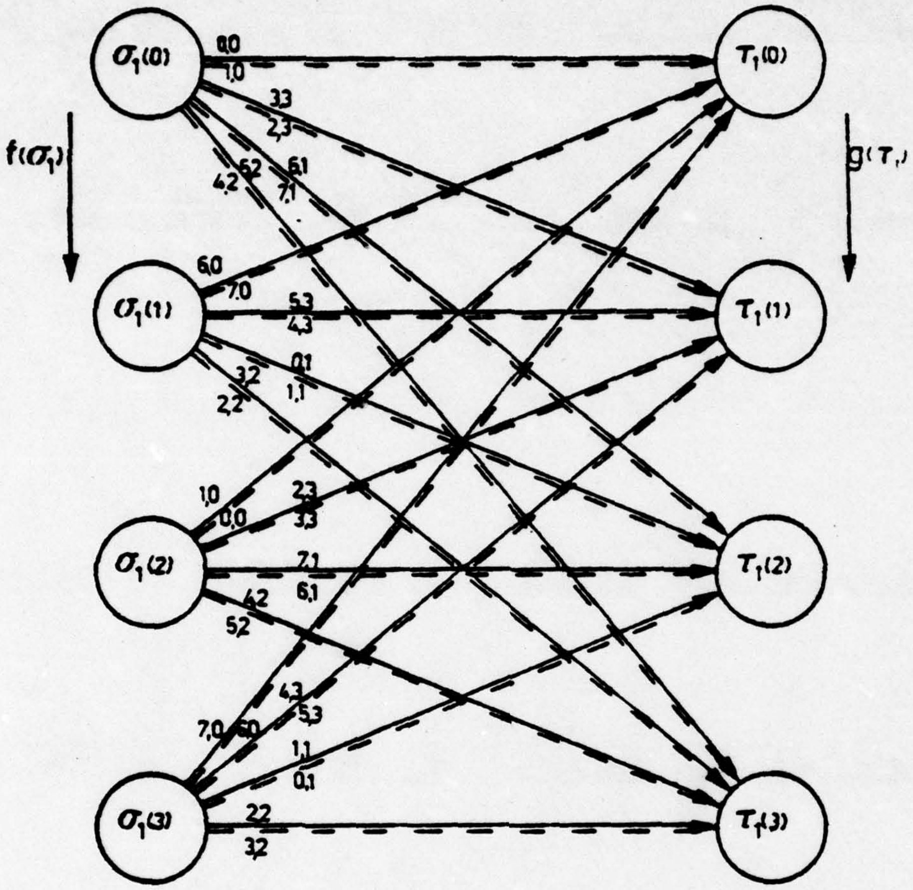


Fig. 9. The t-th section of the trellis diagram, t=...,-1,0,+1,... .

where $\bar{\omega}$ is the modulo 2 complement of ω. Note that for each value $\omega=0$ or $\omega=1$ four arrows impinge on each image τ_1. The preimage σ_1 associated with the minimum within the relevant pair of square brackets in (5) is the survivor. The case where we have more candidates for survivor among the preimages will be considered shortly.

In the classical implementation of the Viterbi algorithm [6] each state $\tau_1(j)$, $j=0,1,2,3$, has a metric register MR_j and a path register PR_j associated with it. The metric register is used to store the current value $f_t[\tau_1(j)]$, t=...,-1,0,+1,... , of the metric function. As only the differences between the values of the metric function matter in the decoding algorithm

$$\min_{0 \leq j \leq 3} \{f_t[\tau_1(j)]\}$$

is subtracted from the contents of all metric registers, thus bounding the value of the contents of the metric registers. The path register PR_j stores the sequence of survivors leading up to state $\tau_1(j)$ at time t. The survivor sequence is stored in terms of the associated message vector corrections ...,$[\hat{e}_{m1,t-1},\ldots,\hat{e}_{mk,t-1}]$, $[\hat{e}_{m1,t},\ldots,\hat{e}_{mk,t}]$; t=...,-1,0,+1,... . Observe that all quantities that are crucial to the estimation algorithm that determines $\hat{\underline{e}}_m(X)$ given $\omega(X)$ are contained in the trellis section of Fig. 9.

Now observe that (5b) and (5d) are identical. Hence, the states $\tau_1(1)$ and $\tau_1(3)$ have identical metric register contents. Moreover, selecting the identical survivor σ_1 in case of a tie, $\tau_1(1)$ and $\tau_1(3)$ also have the same path register contents. As far as metric register and path register contents are concerned, the states $\tau_1(1)$ and $\tau_1(3)$ are not distinct. The metric register and the path register of either state $\tau_1(1)$ or state $\tau_1(3)$ can be eliminated. Apparently, certain symmetries in the state space of the syndrome former can be exploited to reduce the amount of decoder hardware! In the next two sections we further explore this possibility of reducing decoder hardware by introducing certain symmetries in the state space.

For further details on the implementation of the syndrome decoder one is referred to [3]. In this same paper Schalkwijk and Vinck also suggest a slightly modified decoder implementation that uses a read only memory (ROM) thus eliminating the need for metric registers altogether.

IV. SPECIAL R=(n-1)/n CODES-METRIC/PATH REGISTER SAVINGS

Without further ado we now introduce the class $\Gamma_{n,h,\ell}$ of rate-(n-1)/n binary convolutional codes (A,B,C,...,D), i.e. in terms of their syndrome formers, that exhibits state space symmetries that allow for an exponential reduction of decoder hardware. To wit (A,B,C,...,D) $\in \Gamma_{n,h,\ell}$ if and only if

$$a_h = 1 \tag{6a}$$

$$a_j = b_j \ ; \ 0 \leq j \leq \ell-1 \tag{6b}$$

$$a_j = b_j \ ; \ h-\ell+1 \leq j \leq h \tag{6c}$$

$$C,\ldots,D \text{ all have degree } \leq h-\ell \tag{6d}$$

$$\gcd(A,B,C,\ldots,D) = 1 \tag{6e}$$

$$L[\varepsilon_1,(\alpha+\beta)_0,\gamma_0,\ldots,\delta_0] \cap L[(\alpha+\beta)_1,\ldots,(\alpha+\beta)_{\ell-1}] = \{\underline{0}\} \tag{6f}$$

Note that the code $A(X) = 1+X+X^2$, $B(X) = 1+X^2$, $C(X) = 1$ of Fig. 2 is an element of $\Gamma_{3,2,1}$. The code $A(X) = 1+X+X^2+X^4$, $B(X) = 1+X+X^4$ of Fig. 7 is an element of $\Gamma_{2,4,2}$. As a consequence of (6) we have

$$\Gamma_{n,h,1} \supset \Gamma_{n,h,2} \supset \Gamma_{n,h,3} \supset \cdots . \tag{7}$$

If condition (6e) is satisfied, then it follows from the invariant factor theorem [4] that the n-tuple (A,B,C,...,D) is a set of syndrome polynomials for some non-catastrophic rate-(n-1)/n convolutional code (in fact, for a class of such codes).

Assume $\Gamma_{n,h,\ell} \neq \phi$. For (A,B,C,...,D) $\in \Gamma_{n,h,\ell}$ an "ℓ-singleton state" is defined to be a state the last ℓ components of which vanish. Linear combinations and left shifts of ℓ-singleton states are ℓ-singleton states, too. For every state ϕ_1 the states $\phi_i (i \geq \ell+1)$ are ℓ-singleton states. We have the following lemma, the proof of which is left to the reader.

LEMMA 1: For every state σ_1 there exists a unique ℓ-singleton state $\phi_{\ell+1}$ and a unique index set $I \subset \{1,2,\ldots,\ell\}$ such that

$$\sigma_1 = \phi_{\ell+1} + \sum_{i \in I} \alpha_i \, . \tag{8}$$

Using this lemma we now associate with the state σ_1 the set $[\sigma_1]^{(\ell)}$ defined by

$$[\sigma_1]^{(\ell)} = \{\phi_{\ell+1} + \sum_{i \in I} [\alpha_i + r_i(\alpha+\beta)_i] \mid r_i \in \{0,1\} \text{ for all } i\} \, .$$

We shall prove the following theorem:

THEOREM 2: The collection of all sets $[\sigma_1]^{(\ell)}$ forms a partition of the state space.

PROOF: Obviously the union of all $[\sigma_1]^{(\ell)}$ is equal to the state space. So, we only have to prove that $[\sigma_1]^{(\ell)} = [\sigma_1']^{(\ell)}$ whenever $[\sigma_1]^{(\ell)} \cap [\sigma_1']^{(\ell)} \neq \phi$. Let us assume that $[\sigma_1]^{(\ell)} \cap [\sigma_1']^{(\ell)} \neq \phi$. Then there exist r_i and s_i such that

$$\phi_{\ell+1} + \sum_{i \in I} [\alpha_i + r_i(\alpha+\beta)_i] = \phi'_{\ell+1} + \sum_{i \in I'} [\alpha_i + s_i(\alpha+\beta)_i] \, ,$$

or

$$\phi_{\ell+1} - \phi'_{\ell+1} + \sum_{i \in I} r_i(\alpha+\beta)_i - \sum_{i \in I'} s_i(\alpha+\beta)_i = \sum_{i \in I \Delta I'} \alpha_i \, .$$

Now the LSH of above equation is an ℓ-singleton state by (6b,c) so that the symmetric difference $I \Delta I'$ must be empty, in other words $I=I'$. Therefore we get

$$\phi'_{\ell+1} - \phi_{\ell+1} = \sum_{i \in I} (r_i - s_i)(\alpha+\beta)_i \ ,$$

i.e. $\phi'_{\ell+1}$ and $\phi_{\ell+1}$ differ by some linear combination of $\{(\alpha+\beta)_i | i \in I\}$. But then we must have $[\sigma_1]^{(\ell)} = [\sigma_1']^{(\ell)}$, since in the construction of these classes all linear combinations of $\{(\alpha+\beta)_i | i \in I\}$ are involved.

Q.E.D.

__CORROLLARY__: Based on the partition of the state space according to Theorem 2 an equivalence relation $R_{n,h,\ell}$ can be defined, where two states σ_1 and σ_1' are called $R_{n,h,\ell}$-equivalent iff $[\sigma_1]^{(\ell)} = [\sigma_1']^{(\ell)}$.

The one-element equivalence classes of $R_{n,h,\ell}$ consists of exactly one ℓ-singleton state. An example are the states 0,4,8 and 12 in Fig. 8. The number $N_{n,h,\ell}$ of $R_{n,h,\ell}$-equivalence classes can be found as follows. First, take $I \subset \{1,2,\dots,\ell\}$ in (8) fixed, and let j denote the cardinality of I. The last ℓ components of an ℓ-singleton state are zero. Hence, there are $2^{h-\ell}$ ℓ-singleton states. Now 2^j of these $2^{h-\ell}$ ℓ-singleton states correspond to the same $R_{n,h,\ell}$-equivalence class, i.e. all ℓ-singleton states differing by a linear combination of $\{(\alpha+\beta)_i | i \in I\}$. Hence there are $2^{h-\ell-j}$ $R_{n,h,\ell}$-equivalence classes for each I of cardinality j. Thus

$$N_{n,h,\ell} = \sum_{j=0}^{\ell} \binom{\ell}{j} 2^{h-\ell-j} = 2^{h-2\ell} 3^{\ell} \ . \qquad (9)$$

IV. Special R=(n-1)/n codes-metric/path register savings

THEOREM 3: Let $(A,B,C,\ldots,D) \in \Gamma_{n,h,\ell}$, and assume that $1 \le \ell' \le \ell$. Then every $R_{n,h,\ell}$-equivalence class of $(A,B,C,\ldots,D)$ is a union of $R_{n,h,\ell'}$-equivalence classes of $(A,B,C,\ldots,D)$, cf (7).

PROOF: Let σ_1 and τ_1 be $R_{n,h,\ell'}$-equivalent states of $(A,B,C,\ldots,D)$. Then we may write for some $r_i \in \{0,1\}$, $i \in I' \subset \{1,2,\ldots,\ell'\}$:

$$\sigma_1 = \phi_{\ell'+1} + \sum_{i \in I'} \alpha_i ,$$

$$\tau_1 = \phi_{\ell'+1} + \sum_{i \in I'} [\alpha_i + r_i(\alpha+\beta)_i] .$$

On the other hand, for some $I'' \subset \{\ell'+1,\ldots,\ell\}$ and some $\Psi_{\ell+1}$ we have

$$\phi_{\ell'+1} = \Psi_{\ell+1} + \sum_{i \in I''} \alpha_i .$$

Letting $I = I' \cup I''$, $r_i = 0$ for $i \in I''$ we now obtain

$$\sigma_1 = \Psi_{\ell+1} + \sum_{i \in I} \alpha_i$$

$$\tau_1 = \Psi_{\ell+1} + \sum_{i \in I} [\alpha_i + r_i(\alpha+\beta)_i] ,$$

i.e. σ_1 and τ_1 are $R_{n,h,\ell}$-equivalent Q.E.D.

In Fig. 8 we exhibit the $R_{2,4,2,}$-equivalence classes for the $A(X) = 1+X+X^2+X^4$, $B(X) = 1+X+X^4$ code. We claimed that any two states within the same equivalence class have the same metric value irrespective of the received data vector sequence $\underline{r}(X)$. We are now ready to prove this result.

IV. Special $R=(n-1)/n$ codes-metric/path register savings

<u>THEOREM 4</u>: Assume that $(A,B,C,\ldots,C) \in \Gamma_{n,h,\ell}$. Let f_0 be any starting metric function, and let $\omega_1, \omega_2, \omega_3, \ldots$ be any syndrome sequence. Then every iterate f_u is constant on the $R_{n,h,u}$-equivalence classes of $(A,B,C,\ldots,D)$, $1 \leq u \leq \ell$.

<u>PROOF</u>: The proof is by induction on u. Consider the two $R_{n,h,1}$-equivalent states $\phi_2+\alpha_1$, and $\phi_2+\beta_1$. Obviously they belong to the same sink-tuple. We list their preimages, corresponding noise vectors and syndrome digits according to (3).

Preimage	$\phi_2+\alpha_1$	$\phi_2+\beta_1$
	Noise;Syndrome	Noise;Syndrome
$\phi_1 \quad +z\gamma_0+\ldots+t\delta_0$	$[1,0,z,\ldots,t]^T;\omega_1$	$[0,1,z,\ldots,t]^T;\omega_1$
$\phi_1 \quad +(\alpha+\beta)_0+z\gamma_0+\ldots+t\delta_0$	$[0,1,z,\ldots,t]^T;\omega_1$	$[1,0,z,\ldots,t]^T;\omega_1$
$\phi_1+\epsilon_1 \quad +z\gamma_0+\ldots+t\delta_0$	$[1,0,z,\ldots,t]^T;\bar{\omega}_1$	$[0,1,z,\ldots,t]^T;\bar{\omega}_1$
$\phi_1+\epsilon_1+(\alpha+\beta)_0+z\gamma_0+\ldots+t\delta_0$	$[0,1,z,\ldots,t]^T;\bar{\omega}_1$	$[1,0,z,\ldots,t]^T;\bar{\omega}_1$

We see that on every line, i.e. for every preimage the syndrome bits and the Hamming weights of the state transitions to $\phi_2+\alpha_1$, and $\phi_2+\beta_1$ are identical. Hence, $f_1(\phi_2+\alpha_1) = f_1(\phi_2+\beta_1)$ for every f_0 and every ω_1. This proves the assertion for $u=1$. Now let us assume that the statement is true for a fixed u, $1 \leq u \leq \ell-1$. Let f_0 be any starting metric function and let $\omega_1,\omega_2,\omega_3,\ldots$ be any syndrome sequence. Then, by our induction hypothesis, f_u is constant on the $R_{n,h,u}$-equivalence classes. Let X_1 and X_1' be any pair of $R_{n,h,u}$-equivalent states. Then there is a state Ψ_{u+1} and an index set $I \subset \{1,2,\ldots,u\}$ such that for some $r_i \in \{0,1\}$

$$X_1 = \Psi_{u+1} + \sum_{i \in I} \alpha_i \ ,$$

$$X_1' = \Psi_{u+1} + \sum_{i \in I} [\alpha_i + r_i(\alpha+\beta)_i] \ .$$

IV. Special R=(n-1)/n codes-metric/path register savings

We now consider the cosets S and S' of $L[\varepsilon_1,(\alpha+\beta)_0,\gamma_0,\ldots,\delta_0]$ to which χ_1 and χ_1' belong, respectively, and compare them element wise. The states

$$\chi_1 + p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \ldots + s\delta_0,$$

and

$$\chi_1' + p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \ldots + s\delta_0$$

are obviously $R_{n,h,u}$-equivalent for all $p,q,r,\ldots,s \in \{0,1\}$, since by the definition of ε_1 and by (6c,d) the last u components of $p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \ldots + s\delta_0$ vanish. Furthermore, by (6b) we have

$$\sum_{i\in I} a_i = \sum_{i\in I} [a_i+r_i(a_i+b_i)] .$$

Hence, by (3) the preimages

$$\chi_1 + p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \ldots + s\delta_0,$$

and

$$\chi_1' + p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \ldots + s\delta_0$$

give rise to identical syndrome digits in response to an input vector $[x,y,z,\ldots,t]^T$. These arguments together, however, imply that the values of f_{u+1} on the corresponding state transition images are equal and, hence, f_{u+1} is constant on the $R_{n,h,u+1}$-equivalence classes of $(A,B,C,\dots,D)$. Q.E.D.

Theorem 4 proves that one needs only one metric register for each $R_{n,h,\ell}$-equivalence class. We will now show that, except for the last $\ell-1$ stages, the same is true for the path registers. Let $(A,B,C,\dots,D) \in \Gamma_{n,h,\ell}$. Condition (6f), where $\{(\alpha+\beta)_1,(\alpha+\beta)_2,\dots,(\alpha+\beta)_{\ell-1}\} = \{\underline{0}\}$ for $\ell=1$, implies that a coset of $L[\varepsilon_1,(\alpha+\beta)_0,\gamma_0,\dots,\delta_0]$ and a coset of $L[(\alpha+\beta)_1,(\alpha+\beta)_2,\dots,(\alpha+\beta)_{\ell-1}]$ can have at most one element in common, i.e.

LEMMA 5: No two distinct $R_{n,h,\ell-1}$-equivalent states can belong to the same source tuple.

On the other hand, from the proof of Theorem 4 it follows that whenever X_1 and X_1' are $R_{n,h,\ell-1}$-equivalent, then the same holds for the states

$$X_1 + p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \dots + s\delta_0 \quad ,$$

and

$$X_1' + p\varepsilon_1 + q(\alpha+\beta)_0 + r\gamma_0 + \dots + s\delta_0 \quad , \quad p,q,r,\dots,s \in \{0,1\} \quad ,$$

that form the source-tuples containing X_1 and X_1'. These results lead to a natural equivalence between source-tuples. Two source-tuples are said to be equivalent if they contain a pair of $R_{n,h,\ell-1}$-equivalent states. It is left to the reader to prove that this relation is an equivalence relation. The unique and natural one-to-one correspondence between the states of two equivalent source-tuples, that is induced by the intersection with $R_{n,h,\ell-1}$-equivalence classes is, by the proof of Theorem 4, consistent with the algebraic difference structure of the source-tuples. Hence, in view of Theorem 4, we see that for the m-th iterate f_m, $m \geq \ell-1$, of any metric function f_0 under any syndrome sequence $\omega_1, \omega_2, \omega_3, \ldots$ the values of f_m on the corresponding states of two equivalent source-tuples are identical.

Given two successive iterates f_{j-1} and f_j, $j \geq \ell$, of a metric function f_0, linked by the syndrome digit ω_j,

$$f_{j-1} \xmapsto{\omega_j} f_j \ .$$

In Viterbi decoding [6] one determines for each state τ_1 a survivor σ_1, such that

$$f_j(\tau_1) = f_{j-1}(\sigma_1) + W_H([x,y,z,\ldots,t]^T) \ ,$$

subject to (4). Survivors of a state τ_1 in the sink-tuple T_i always belong to the corresponding source-tuple S_i, see Section II. However, as discussed in Section III, there are situations in which more than one survivor may be chosen, i.e. when two or more σ_1's in (4) achieve the minimum. In this case, one has a choice of two possible strategies that result in the same decoded error rate by transmission over a binary symmetric channel (BSC), i.e. (i) flip a (multi) coin, or (ii) decide for every tie-pattern once and for ever which survivor shall be taken. We shall use the second strategy, that according to the properties of equivalent source-tuples can be realized in the following way: Whenever two source-tuples S_i and S_i' are equivalent (and, hence, have statewise identical f_{j-1}-values) then let for the respective sink-tuples T_i and T_1' statewise corresponding survivors be chosen in such a way, that $R_{n,h,1}$-equivalent states get the same survivor. Given a sequence of metric function iterates

$$f_0 \xmapsto{\omega_1} f_1 \xmapsto{\omega_2} f_2 \longmapsto \dots \longmapsto f_{j-2} \xmapsto{\omega_{j-1}} f_{j-1} \xmapsto{\omega_j} f_j \ , \ j \geq \ell \ ,$$

then for every state σ_1 a sequence of successive survivors can be constructed

$$\sigma_1^{(-j)} \longmapsfrom \sigma_1^{(-j+1)} \longmapsfrom \sigma_1^{(-j+2)} \longmapsfrom \dots \longmapsfrom \sigma_1^{(-2)} \longmapsfrom \sigma_1^{(-1)} \longmapsfrom \sigma_1 \ ,$$

and the following theorem holds.

<u>THEOREM 5</u>: If σ_1 and η_1 are distinct $R_{n,h,m}$-equivalent states then $\sigma_1^{(-m)} = \eta_1^{(-m)}$, $m=1,2,\dots,\ell$.

<u>PROOF</u>: The proof is by induction on m. For m=1 the assertion is part of our assumption above. Now assume that the statement is true for m=u, u fixed, $1 \leq u \leq \ell-1$, and let σ_1 and η_1 be two $R_{n,h,u+1}$-equivalent states, that are not $R_{n,h,u}$-equivalent (otherwise $\sigma_1^{(-u)} = \eta_1^{(-u)}$ and, hence, immediately $\sigma_1^{(-u-1)} = \eta_1^{(-u-1)}$). Then we may write

$$\sigma_1 = \phi_{u+2} + \alpha_{u+1} + \sum_{i \in I} \alpha_i$$

$$\eta_1 = \phi_{u+2} + \beta_{u+1} + \sum_{i \in I} [\alpha_i + r_i (\alpha+\beta)_i] \ ,$$

where $I \subset \{1,2,\dots,u\}$. It is easy to find preimages $\tilde{\sigma}_1$ and $\tilde{\eta}_1$ of σ_1 and η_1 respectively, viz.

$$\tilde{\sigma}_1 = \phi_{u+1} + \alpha_u + \sum_{i\in I\setminus\{1\}} \alpha_{i-1} ,$$

$$\tilde{\eta}_1 = \phi_{u+1} + \beta_u + \sum_{i\in I\setminus\{1\}} [\alpha_{i-1}+r_i(\alpha+\beta)_{i-1}] .$$

Obviously, $\tilde{\sigma}_1$ and $\tilde{\eta}_1$ are $R_{n,h,u}$-equivalent and, hence, by Theorem 3, also $R_{n,h,\ell-1}$-equivalent. Therefore, the source-tuples containing $\tilde{\sigma}_1$ and $\tilde{\eta}_1$ are equivalent. Furthermore, we observe that

$$\sigma_1 = \tilde{\sigma}_2 + \begin{cases} 0 & \text{if} \quad i\notin I \\ \alpha_1 & \text{if} \quad i\in I \end{cases} ,$$

$$\eta_1 = \tilde{\eta}_2 + \begin{cases} 0 & \text{if } i\notin I \\ \alpha_1+r_1(\alpha+\beta)_1 & \text{if } i\in I \end{cases} .$$

Hence, because of the assumption made above, the survivors $\sigma_1^{(-1)}$ and $\eta_1^{(-1)}$ are corresponding states, i.e. $R_{n,h,\ell-1}$-equivalent states. The algebraic difference structure of equivalent source-tuples is identical, hence,

$$\tilde{\sigma}_1 - \sigma_1^{(-1)} = \tilde{\eta}_1 - \eta_1^{(-1)} \in L[\varepsilon_1,(\alpha+\beta)_0,\gamma_0,\ldots,\delta_0] .$$

So, $\tilde{\sigma}_1 - \sigma_1^{(-1)} = \tilde{\eta}_1 - \eta_1^{(-1)}$ is a u-singleton state. Hence, $\sigma_1^{(-1)}$ and $\eta_1^{(-1)}$ are $R_{n,h,u}$-equivalent and therefore, by the induction hypothesis, $\sigma_1^{(-u-1)} = \eta_1^{(-u-1)}$. Q.E.D.

Theorem 5 shows that except perhaps for the last ℓ-1 stages, $R_{n,h,\ell}$-equivalent states have the same path register contents irrespective of the received data vector sequence $\underline{r}(X)$. Thus, roughly, speaking, one needs only one path register for each $R_{n,h,\ell}$-equivalence class of states. By Theorem 4 one only needs one metric register per $R_{n,h,\ell}$-equivalence class. Hence, the complexity [3] of a syndrome decoder for a code $(A,B,C,\ldots,D) \epsilon \Gamma_{n,h,\ell}$ is proportional to the number $N_{n,h,\ell}$ of $R_{n,h,\ell}$-equivalence classes, i.e. by (9) the complexity is proportional to $2^{h-2\ell}3^{\ell}$. As an example take a code in $\Gamma_{2,2\ell,\ell}$, i.e. a rate-½ code with

$$A(X) = X^{2\ell} + A_{2\ell-1}X^{2\ell-1} + \ldots + A_1X + 1 \ ,$$
$$B(X) = A(X) + X^{\ell} \ .$$

The syndrome decoder for such a code has complexity proportional to $3^{\ell} = (\sqrt{3})^h$. The classical Viterbi decoder [6] for the same code has complexity 2^h, hence, by exploiting the state space symmetry we achieve an exponential saving in hardware.

Before extending our present results to rate-k/n codes one comment concerning the free distance of codes $(A,B,C,\ldots,D) \epsilon \Gamma_{n,h,\ell}$ is in order. It is quite obvious that constraints like (6) can reduce the maximum obtainable free distance for given n, and h. We are not yet able to derive a lower bound on the free distance of codes $(A,B,C,\ldots D) \epsilon \Gamma_{n,h,\ell}$. However, Table I of the next section lists the free distance of some short constraint length codes in $\Gamma_{n,h,\ell}$. It turns out that at least for these constraint lengths the free distance for the codes satisfying the constraints (6) is very close to the maximum achievable free distance for the given values of n, and h.

V. SPECIAL R=k/n CODES-METRIC/PATH REGISTER SAVINGS

The syndrome former of a rate-k/n convolutional code consists of n-k syndrome formers of the type considered in Section II, all sharing the same set of nh memory cells, compare Fig. 4. Hence, the n-k syndrome formers in the set $\{\Sigma^1,\Sigma^2,\ldots,\Sigma^{n-k}\}$, where $\Sigma^i \triangleq (A_i,B_i,C_i,\ldots,D_i)$, all have the same physical state, i.e. the contents of the nh memory cells they have in common. To obtain the metric/path register savings that were realized in Section IV each of the syndrome formers Σ^i, $i=1,2,\ldots,n-k$, should be in $\Gamma_{n,h,\ell}$, and the common physical states should have the same equivalence classes w.r.t. the equivalence relation of syndrome-indistinguishability in each of the n-k individual syndrome formers. We will call a set of rate-(n-1)/n syndrome formers that share a common physical state "coherent" if the individual syndrome formers have the same abstract states.

Let $\Gamma^{(n-k)}_{n,h,\ell}$ be the class of codes that are defined by n-k coherent syndrome formers each of which is in $\Gamma_{n,h,\ell}$. Table I lists the maximum free distance for various values of the parameters k,n,h, and ℓ. The $\Gamma^{(n-k)}_{n,h,\ell}$ classes with (k,n) = (1,3) are defined by two coherent syndrome formers. The column with "N" on top gives the maximum free distance for the relevant values of k,n, and h dropping the coherence requirement. Comparing the N-column with the ℓ=1-column both for (k,n) = = (1,3) gives some idea of the effect of the coherence requirement on the free distance. Table II lists several optimal $\Gamma^{(n-k)}_{n,2\ell,\ell}$ codes in terms of their syndrome former connections, geometrically arranged as in Fig. 4.

V. Special R=k/n codes-metric/path register savings

TABLE I

MAXIMUM FREE DISTANCE OF VARIOUS $\Gamma^{(n-k)}_{n,h,\ell}$ - CLASSES

(k,n)	(k,n) = (1,2)				(k,n) = (2,3)				(k,n) = (1,3)				
$h \backslash \ell$	1	2	3	4	1	2	3	4	N	1	2	3	4
2	5				3				8	7			
3	6				4				10	9			
4	7	7			5	5			12	11	10		
5	8	8			6	6			13	12	12		
6	10	9	8		6	6	6		15	14	14	13	
7	10	10	10		8	7	6		16	16	16	15	
8	12	11	10	10	8	8	8	6	18	18	17	16	16
9	12	12	12	11	8	8	8	8	20	20	19	18	18

TABLE II

OPTIMAL $\Gamma^{(n-k)}_{n,2\ell,\ell}$ - CODES

$\ell \backslash$ (k,n)	(k,n) = (1,2)	(k,n) = (2,3)	(k,n) = (1,3)	
	Σ	Σ	Σ^1	Σ^2
1	5,7	5,7,1	5,7,0	6,4,1
2	23,27	23,27,5	37,33,0	32,36,1
3	107,117	103,113,7	133,123,0	124,134,1
4	453,473	403,423,7	453,473,0	464,444,1

The remainder of this section will be devoted to a study of the newly defined concept of coherence of syndrome formers. Consider two syndrome formers

$$\Sigma \overset{\Delta}{=} (A,B,C,\dots,D) \ ,$$

and

$$\Sigma' \overset{\Delta}{=} (A',B',C',\dots,D')$$

sharing the same set of nh memory cells, compare Fig. 4. From the mathematical point of view the syndrome-indistinguishability classes of a syndrome former Σ can be considered as cosets of the set of those physical states that have an all zero syndrome sequence in response to a sequence of all zero noise vectors. Hence, we may state that Σ and Σ' are coherent if and only if for all nh-tuples

$$(x_1,\dots,x_h;y_1,\dots,y_h;z_1,\dots z_h;\dots;t_1,\dots,t_h)$$

we have

$$\sum_{i=1}^{h} (x_i\alpha_{h+1-i} + y_i\beta_{h+1-i} + \dots + t_i\delta_{h+1-i}) = \underline{0} \Longleftrightarrow$$

$$\Longleftrightarrow \sum_{i=1}^{h} (x_i\alpha'_{h+1-i} + y_i\beta'_{h+1-i} + \dots + t_i\delta'_{h+1-i}) = \underline{0} \ .$$

We shall now discuss some consequences of this concept of coherence.

1/ Let Σ and Σ' be coherent syndrome formers, and assume as before that $a_h = 1$. Then $\{\alpha_1,\alpha_2,\dots,\alpha_h\}$ is a basis for the abstract state space of Σ. In other words

$$\sum_{i=1}^{h} x_i \alpha_{h+1-i} = \underline{0} \Leftrightarrow x_1 = x_2 = \ldots = x_h = 0$$

so that by coherence

$$\sum_{i=1}^{h} x_i \alpha'_{h+1-i} = \underline{0} \Leftrightarrow x_1 = x_2 = \ldots = x_h = 0 \ .$$

Hence, $\{\alpha'_1, \alpha'_2, \ldots, \alpha'_h\}$ is a basis for the abstract state space of Σ' and $a'_h = 1$.

2/ Let Σ and Σ' be coherent syndrome formers, $a_h = a'_h = 1$.

Then the correspondence

$$\sum_{i=1}^{h} (x_i \alpha_{h+1-i} + \ldots + t_i \delta_{h+1-i}) \Leftrightarrow \sum_{i=1}^{h} (x_i \alpha'_{h+1-i} + \ldots + t_i \delta'_{h+1-i})$$

is an isomorphism between the abstract state spaces of Σ and Σ'.

<u>SKETCH OF PROOF</u>: By 1/, $\{\alpha_1, \alpha_2, \ldots, \alpha_h\}$ and $\{\alpha'_1, \alpha'_2, \ldots, \alpha'_h\}$ are bases of the state spaces above. Hence, for example

$$\beta_1 = u_1\alpha_1 + u_2\alpha_2 + \ldots + u_h\alpha_h \ ,$$

or

$$-\beta_1 + u_1\alpha_1 + u_2\alpha_2 + \ldots + u_h\alpha_h = \underline{0} \ ,$$

so that by coherence

$$-\beta'_1 + u_1\alpha'_1 + u_2\alpha'_2 + \ldots + u_h\alpha'_h = \underline{0},$$

i.e.

$$\beta'_1 = u_1\alpha'_1 + u_2\alpha'_2 + \ldots + u_h\alpha'_h \ , \qquad \text{etc.}$$

3/ Let Σ and Σ' be coherent syndrome formers, i.e. $a_h = a_h' = 1$. Then Σ and Σ' have isomorphic source/sink-tuple structures.

PROOF: Sink-tuples in the state space of Σ are cosets of $L[\alpha_1,\beta_1,\gamma_1,\dots,\delta_1]$, and this subspace corresponds by 2/, in the obvious way, by coherence, to $L[\alpha_1',\beta_1',\gamma_1',\dots,\delta_1']$. Source-tuples in the state space of Σ are cosets of the set S of those abstract states that have image $\underline{0}$ under state transition. Let $\sigma_1 \triangleq \sum_{i=1}^{h} u_i\alpha_i$ such that $\sigma_1 \mapsto \underline{0}$ under state transition with the noise vector $[x,y,z,\dots,t]^T$. Then we have

$$\sum_{i=1}^{h-1} u_i\alpha_{i+1} + x\alpha_1 + y\beta_1 + z\gamma_1 + \dots + t\delta_1 = \underline{0} \ ,$$

so that by coherence

$$\sum_{i=1}^{h-1} u_i\alpha_{i+1}' + x\alpha_1' + y\beta_1' + z\gamma_1' + \dots + t\delta_1' = \underline{0} \ ,$$

which means that in the state space of Σ', when we define $\sigma_1' \triangleq$ $\triangleq \sum_{i=1}^{h} u_i\alpha_i'$, also $\sigma_1' \mapsto \underline{0}$ under state transition with noise vector $[x,y,z,\dots,t]^T$, and vice versa. Hence, coherence implies that both

$$L[\alpha_1,\beta_1,\gamma_1,\dots,\delta_1] \leftrightarrow L[\alpha_1',\beta_1',\gamma_1',\dots,\delta_1'] \ ,$$

and

$$S \leftrightarrow S'$$

by the isomorphism defined in 2/. This implies that also the cosets of $L[\alpha_1,\beta_1,\gamma_1,\dots,\delta_1]$ and S, and the cosets of $L[\alpha_1',\beta_1',\gamma_1',\dots,\delta_1']$ and S' have isomorphic intersections. Q.E.D.

4/ Finally, we can restate the coherence of Σ and Σ' in terms of a condition of their polynomials A,B,C,...,D , and A',B',C',...,D' as follows. Let Σ and Σ' be coherent syndrome formers, $a_h = a'_h = 1$. Let the isomorphism between their state spaces, which is generated by the mapping $\alpha_j \mapsto \alpha'_j$, $j = h,h-1,\ldots,1$, w.r.t. the natural basis of unit vectors be given by the (invertible) matrix Q, i.e.

$$Q \begin{bmatrix} 1 & 0 & 0 & . & 0 \\ a_{h-1} & 1 & 0 & . & 0 \\ . & . & . & . & . \\ a_1 & a_2 & a_3 & . & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & . & 0 \\ a'_{h-1} & 1 & 0 & . & 0 \\ . & . & . & . & . \\ a'_1 & a'_2 & a'_3 & . & 1 \end{bmatrix} . \tag{10}$$

It is immediately verified that Q itself has the form

$$Q = \begin{bmatrix} 1 & 0 & 0 & . & 0 \\ q_{h-1} & 1 & 0 & . & 0 \\ . & . & . & . & . \\ q_1 & q_2 & q_3 & . & 1 \end{bmatrix} .$$

The matrix identity (10) can be reformulated as a polynomial congruence, i.e.

$$\left(\sum_{i=0}^{h-1} q_{h-i}\, X^i\right) \left(\sum_{i=0}^{h-1} a_{h-i}\, X^i\right) \equiv \sum_{i=0}^{h-1} a'_{h-i}\, X^i \,(\mathrm{mod}\ X^h) \ , \ q_h \triangleq 1 \ .$$

The isomorphism also entails that

$$\left(\sum_{i=0}^{h-1} q_{h-i}\, X^i\right) \left(\sum_{i=0}^{h-1} b_{h-i}\, X^i\right) \equiv \sum_{i=0}^{h-1} b'_{h-i}\, X^i \,(\mathrm{mod}\ X^h) \ , \text{ etc.}$$

Elimination of $\sum_{i=0}^{h-1} q_{h-i}X^i$ yields

$$\left(\sum_{i=0}^{h-1} a'_{h-i}X^i\right)\left(\sum_{i=0}^{h-1} b_{h-i}X^i\right) - \left(\sum_{i=0}^{h-1} a_{h-i}X^i\right)\left(\sum_{i=0}^{h-1} b'_{h-i}X^i\right) \equiv$$

$$\equiv 0 \pmod{X^h} .$$

Reversing the order of the coefficients in the polynomials of this congruence we find

$$\text{degree}\left[\sum_{i=0}^{h-1} a'_{i+1}X^i \sum_{i=0}^{h-1} b_{i+1}X^i - \sum_{i=0}^{h-1} a_{i+1}X^i \sum_{i=0}^{h-1} b'_{i+1}X^i\right] \le h-2 ,$$

or

$$\text{degree } [A'(X)B(X) - B'(X)A(X)] \le h, \text{ etc.}$$

In fact, this reasoning can also be given in the opposite direction, where we construct, given $a_h = a'_h = 1$, the polynomial $\tilde{Q}(X)$ and, hence, the transformation Q as

$$\tilde{Q}(X) \triangleq \sum_{i=0}^{h-i} q_{h-i}X^i = \left(\sum_{i=0}^{h-i} a'_{h-i}X^i\right)\left(\sum_{i=0}^{h-i} a_{h-i}X^i\right)^{-1} \bmod X^h .$$

Note that $a_h = 1$ and, hence, the polynomial $\sum_{i=0}^{h-1} a_{h-i}X^i$ is invertible mod X^h . So we have the following theorem.

<u>THEOREM 6</u>: Two syndrome formers $\Sigma \triangleq (A,B,C,\ldots,D)$ and $\Sigma' \triangleq (A',B',C',\ldots,D')$, where $h = h'$, are coherent if and only if all 2x2 subdeterminants of the polynomial matrix

$$\begin{bmatrix} A(X) & B(X) & C(X) & \cdots & D(X) \\ A'(X) & B'(X) & C'(X) & \cdots & D'(X) \end{bmatrix}$$

have degree $\le h$.

We conclude this section with an example. Consider the binary rate-1/3 convolutional code generated by an encoder with connection polynomials $1+X^2+X^5+X^6$, $1+X^2+X^3+X^5+X^6$, and $X^3+X^4+X^5+X^6$. The inverse encoder of minimal degree is unique, and is given by the polynomials $1+X+X^2$, X, and X^2. The free distance of the above code is 13 (the maximum free distance for a rate-1/3 code with polynomials of degree 6 is 15). A set of syndrome formers of minimal degree is given by $1+X+X^3$, $1+X$, $X+X^3$, and X^2 , X^2+X^3 , $1+X+X^3$. The implementation of a decoder using this particular set of syndrome formers requires $2^6 = 64$ metric/path register combinations. The set of syndrome formers $1+X^3+X^6$, $1+X^3$, $1+X+X^4+X^6$, and $1+X+X^2+X^4+X^5+X^6$, $1+X+X^2+X^3$, $X^2+X^5+X^6$ is coherent, but a decoder using this particular set of syndrome formers also requires $2^6 = 64$ metric/path register combinations. The set of syndrome formers $1+X+X^6$, $1+X+X^4+X^6$, $1+X+X^3+X^4$, and $1+X^2+X^5+X^6$, $1+X^2+X^3+X^4+X^5+X^6$, $X+X^2+X^4$ is coherent and is a subset of $\Gamma_{3,6,2}$ and, hence, our code belongs to $\Gamma_{3,6,2}^{(3-1)}$ and therefore, by (9) the corresponding decoder can be implemented with $N_{3,6,2} = 36$ metric/path register combinations!

VI CONCLUSIONS

This paper describes the operation of a syndrome decoder for binary rate-k/n convolutional codes in terms of the state space of its syndrome former. A class $\Gamma_{n,h,\ell}^{(n-k)}$ of convolutional codes is defined that exhibits certain state space symmetries that allow for an exponential reduction of decoder hardware. The maximum free distance of several short constraint length $\Gamma_{n,h,\ell}^{(n-k)}$ classes is listed in Table I. Codes achieving the maximum free distance of several $\Gamma_{n,2\ell,\ell}^{(n-k)}$ classes are given in Table II. These $\Gamma_{n,2\ell,\ell}^{(n-k)}$ classes offer the largest hardware savings!

Presently, we are investigating whether the state space formalism developed in this paper can also be used to advantage in sequential decoding. It will then become interesting to find the maximum free distance of classes of long constraint length codes that exhibit certain state space symmetries. This is one of our present topics of research.

ACKNOWLEDGEMENT

The authors want to thank A.J.P. de Paepe for several useful discussions and Miss G. van Hulsen for the accurate typing of the manuscript.

REFERENCES

[1] J.P.M. Schalkwijk and A.J. Vinck,
"Syndrome decoding of convolutional codes",
IEEE Trans. Commun. (Corresp.), vol. COM-23, pp. 789-792,
July 1975.

[2] J.P.M. Schalkwijk,
"Symmetries of the state diagram of the syndrome former of
a binary rate-½ convolutional code",
Lecture Notes, CISM Udine Summer School on Coding, Udine,
Italy, September 2-12, 1975.

[3] J.P.M. Schalkwijk and A.J. Vinck,
"Syndrome decoding of binary rate-½ convolutional codes",
IEEE Trans. Commun., vol. COM-24, pp. 977-985, September 1976.

[4] G.D. Forney, Jr.,
"Convolutional codes I: Algebraic structure",
IEEE Trans. Inform. Theory, vol. IT-16, pp. 720-738,
November 1970;
also, correction appears in vol. IT-17, p. 360, May 1971.

[5] G.D. Forney, Jr.,
"Structural analysis of convolutional codes via dual codes",
IEEE Trans. Inform. Theory, vol. IT-19, pp. 512-518, July 1973.

References

[6] A.J. Viterbi,

"Convolutional codes and their performance in communication systems",

IEEE Trans. Commun. Technol. (Special Issue on Error Correcting Codes - Part II), vol. COM-19, pp. 751-772, October 1971.

3.0 SOFT DECISION SYNDROME DECODING

Abstract

A method is given for soft decision syndrome decoding. Then the influence of soft decision on the hardware complexity will be discussed.

1. INTRODUCTION

The principle of "Viterbi like" syndrome decoding for binary convolutional codes will be explained using the R=½ binary code generated by the encoder of *Fig. 1.*

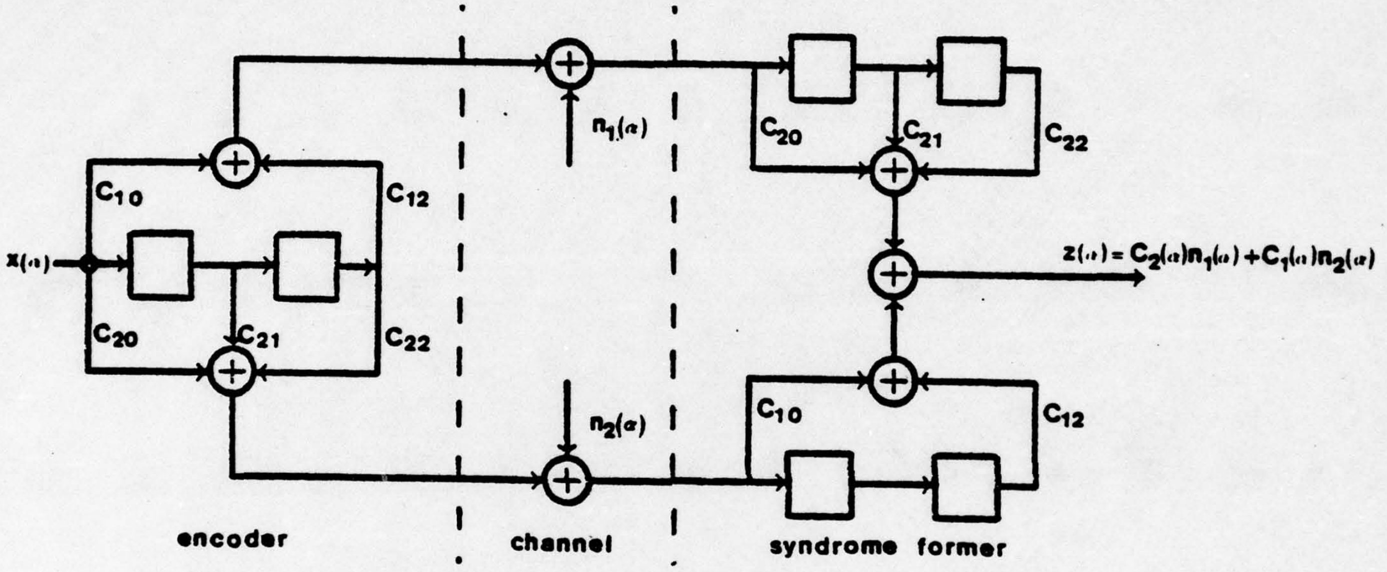


Fig. 1. Encoding and syndrome forming circuit for a R=½ code

The encoder has connection polynomials $C_1(\alpha)$ and $C_2(\alpha)$. Hence, the encoder outputs are $C_1(\alpha)x(\alpha)$ and $C_2(\alpha)x(\alpha)$. The syndrome $z(\alpha)$ only depends on $n_1(\alpha)$ and $n_2(\alpha)$, for

$$\begin{aligned} z(\alpha) &= C_2(\alpha)[C_1(\alpha)x(\alpha) + n_1(\alpha)] + C_1(\alpha)[C_2(\alpha)x(\alpha) + n_2(\alpha)] \\ &= C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha) \end{aligned} \quad (1)$$

For a non catastrophic code, $C_1(\alpha)$ and $C_2(\alpha)$ are relatively prime. Hence, there exist polynomials $D_1(\alpha)$ and $D_2(\alpha)$ such that $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$. The estimate $\hat{x}(\alpha)$ of the data sequence $x(\alpha)$ can be written as

$$\hat{z}(\alpha) = [D_1(\alpha)y_1(\alpha) + D_2(\alpha)y_2(\alpha)] + \hat{\omega}(\alpha) \tag{2}$$

where

$$\hat{\omega}(\alpha) = D_1(\alpha)\hat{n}_1(\alpha) + D_2(\alpha)\hat{n}_2(\alpha) \tag{3}$$

and

$$y_i(\alpha) = C_i(\alpha)x(\alpha) + n_i(\alpha) \; ; \; i = 1,2 \tag{4}$$

According to [1], we can draw the state diagram, see *Fig. 2*, for the syndrome forming circuit of *Fig. 1*. Solid transitions in *Fig. 2* correspond to $z_k = 0$, and dashed transitions to $z_k = 1$, $k = \cdots,-1,0,1,\cdots$. Next to each transition one finds the values of $\hat{n}_{k1}, \hat{n}_{k2}; \hat{\omega}_k$, $k = \cdots,-1,0,1,\cdots$. With each state in *Fig. 2*, we associate a metric register, and a path register. The metric register contains the logarithm of the path likelihood of the most likeli path entering a particular state $s_j(k)$, $j = 0,1,\cdots,2^{\nu-1}$ at the particular time k, $k = \cdots,-1,0,1,\cdots$. The logarithm of the likelihood is taken, for then the new metric at time k+1 is the sum of the old metric at time k and the branch metric. The coefficients $\hat{\omega}^j_{k-D+1}, \hat{\omega}^j_{k-D+2}, \cdots, \hat{\omega}^j_k$

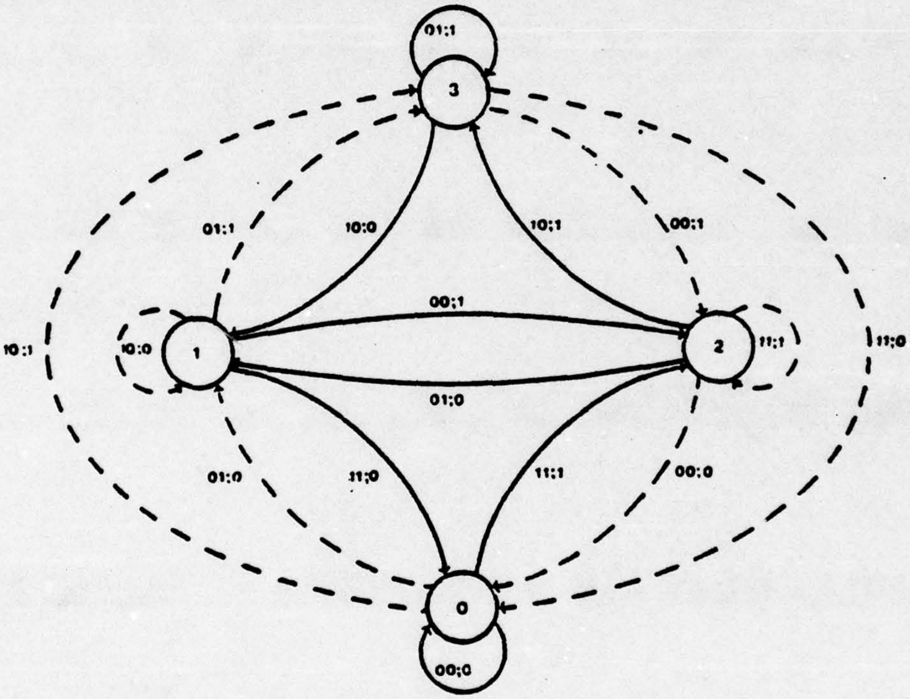


Fig. 2. State diagram of the syndrome former

$k = \cdots,-1,0,1,\cdots$, associated with the path $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]^j$ of maximum log likelihood are stored in the path register for the j-th state. The oldest bit $\hat{\omega}^j_{k-D+1}$ on the path with largest metric is added to

$$[D_1(\alpha)y_1(\alpha) + D_2(\alpha)y_2(\alpha)]_{k-D+1} \tag{5}$$

to give the estimate $\hat{x}(\alpha)$ of (2).

2. QUANTIZATION OF THE RECEIVED CODE SYMBOLS

As pointed out in [2], 180° *binary phase shift keying* (BPSK) in combination with coding is an efficient way of communication over Gaussian channels. Quantization of the demodulated received code symbols, is to facilitate digital processing at the decoder. When 8-level quantization is used, about 0.25 dB in received signal to noise ratio is lost, compared with infinitely fine quantization. Hence, further quantization is questionable. With 2-level (binary) quantization the loss in SNR is roughly 2 dB. *Fig. 3* shows the quantization schemes for 2, 4 and 8 levels, where +1 corresponds with a code symbol 1, and -1 with a code symbol 0.

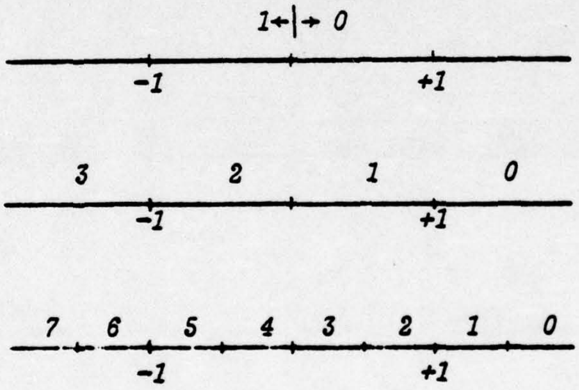


Fig. 3. Quantization scheme for 2, 4 and 8 levels

The spacings in the above schemes can be shown to be almost optimum. The Gaussian channel with modulator and demodulator is then equivalent to a discrete channel with two inputs, and 2, 4 or 8 outputs, respectively. The channel transition probabilities are equal to the probabilities that a $\sqrt{\frac{N_0}{2Es}}$ variance Gaussian random variable with variance $\sqrt{\frac{N_0}{2Es}}$ and mean $\pm$ 1 lies in the intervals indicated in *Fig. 3*. The problem we are now faced with is the adjustment of the syndrome decoder. Take, for example, a 4-level quantizer as indicated in *Fig. 4*.

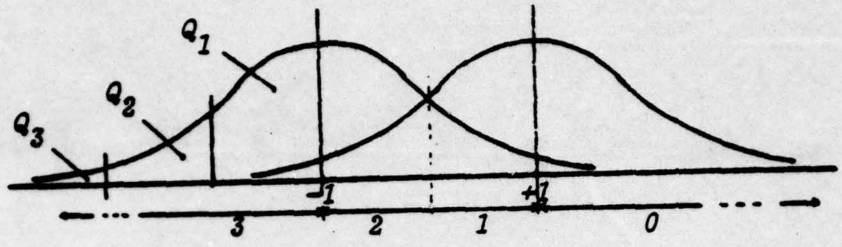


Fig. 4. Probability density function of the received signal

Let a received signal lie in interval 2. The syndrome forming circuit only accepts the symbols 0 and 1. Hence, a binary quantizer is used to set the received signal equal to 0. Now there are two possabilities, the relevant noise bit could either be zero or one with probability Pr(0) = Q_1 and Pr(1) = Q_2, respectively. The same can be said about a received signal lying in interval 1. For the intervals 0 and 3, Pr(0) = ½ and Pr(1) = Q_3. In fact, we only need the absolute value of the received signal to determine Pr(0) and Pr(1) and thus the branch metric. From simulations, it follows that the decoder is quite insensitive to branch metric quantization. Hence, use of integers instead of exact log likelihoods gives a very small performance degradation. *Fig.* 5 shows a possible set of metrics for the case of 4-level quantization.

Hard quantized noise	*Received quantized level*			
	0	*1*	*2*	*3*
0	*0*	*1*	*1*	*0*
1	*3*	*2*	*2*	*3*

Fig. 5. Metric quantization scheme

In [1] is shown that in general, the number of different state metrics for a R = ½ "Viterbi like" syndrome decoder with 2^{ν} states is equal to $\frac{3}{4} \cdot 2^{\nu}$. For a special class of codes, this can even be further reduced to $(\sqrt{3})^{\nu}$. The reduction was mainly based on the fact that the branch metric contribution of the noise pairs [0,1] and [1,0] are equal. From *Fig.* 5 it follows that in the soft decision decoder, the above mentioned contributions are not always equal. For example, let a received signal pair be in the intervals 0 and 1. Then, the branch metric for a [0,1]-branch is 2, while the contribution for a [1,0]-branch is 4. A direct consequence is that the R = ½ syndrome decoder has 2^{ν} different state metrics. Each state having 4 entries, which leads to a more complicated path register reshuffling and metric calculation scheme as compared to the Viterbi decoder.

CONCLUSIONS

A method is given for soft decision "Viterbi like" syndrome decoding. This method is not only valid for $R = \frac{1}{2}$, but for all $R = k/n$ "Viterbi like" syndrome decoders. The number of different state metrics in the $R = \frac{1}{2}$ case cannot be reduced. It seems that the advantages of syndrome decoding for long constraint length codes disappear.

Implementation of a R = ½ convolutional decoder

The three major functions to be perfomed by the encoder are

1/ determination of the branch metrics,

2/ determination of the new state metrics and

3/ generation of the survivor sequence.

The hardware complexity of the Viterbi and of the syndrome decoder strongly depends upon the constraint length $\underline{\nu}$ of the encoder, the code rate, and also if hard (Q = 2) or shoft (Q = 4,8) decision is used on the received data stream. First a general comparison will be made for R = ½ and Q = 2. Then the influence of soft decision will be discussed. For Viterbi decoding the new metric of any state (S) is calculated according to

$$M_S(k+1) = \min \left[\{M_{\alpha S}(k) + d(\underline{c},\underline{r})\} , \{M_{\alpha S+\alpha^{-\nu}}(k) + d(\bar{\underline{c}},\underline{r})\}\right] \qquad (6)$$

where $d(\underline{c},\underline{r})$ denotes the hamming distance between the transition from state $(\alpha^{-1}S)$ to state (S), and the received bit pair $\underline{r}$. Note that $d(\bar{\underline{c}},\underline{r}) = 2 - d(\underline{c},\underline{r})$, whenever the outermost stages of the encoder are connected with the mod-2 adders. The hardware translation of (6) is given in *Fig. 6*. A constraint length ν code, requires a Viterbi decoder with basicly 2^{ν} of these sections.

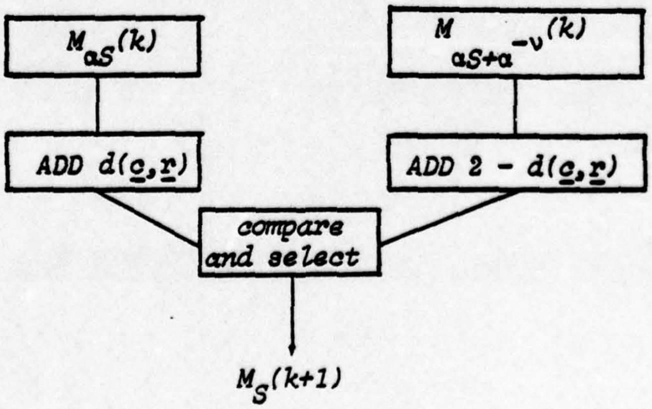

$M_{\alpha S}(k)$
$M_{\alpha S+\alpha^{-\nu}}(k)$
ADD $d(\underline{c},\underline{r})$
ADD $2 - d(\underline{c},\underline{r})$
compare and select
$M_S(k+1)$

Fig. 6. Add and compare logic

In the state table of the syndrome decoder [1], the transitions occur in groups. One group is given in *Fig. 7*, where s_2^b is

state	*transition pair*			
	00	*01*	*11*	*10*
S	$\alpha^{-1}S$	$\alpha^{-1}S+\alpha^{-1}$	$\alpha^{-1}S+s_2^b+\alpha^{-1}$	$\alpha^{-1}S+s_2^b$
$S+(s_2^b+\alpha^{-1})$	$\alpha^{-1}S+s_2^b+\alpha^{-1}$	$\alpha^{-1}S+s_2^b$	$\alpha^{-1}S$	$\alpha^{-1}S+\alpha^{-1}$
$S+\alpha^{-\nu}$	$\alpha^{-1}S$	$\alpha^{-1}S+\alpha^{-1}$	$\alpha^{-1}S+s_2^b+\alpha^{-1}$	$\alpha^{-1}S+s_2^b$
$S+\alpha^{-\nu}+(s_2^b+\alpha^{-1})\alpha$	$\alpha^{-1}S+s_2^b+\alpha^{-1}$	$\alpha^{-1}S+s_2^b+\alpha^{-1}$	$\alpha^{-1}S$	$\alpha^{-1}S+\alpha^{-1}$

Fig. 7. Group of transitions

the base state of the syndrome former. In general, the syndrome decoder has $2^\nu/4$ of these groups. Assume that, without loss of generality, state (S) and state $(S+(s_2^b+\alpha^{-1})\alpha)$ cause the same syndrome outputs. Otherwise interchange (S) and $(S+\alpha^{-\nu})$. The metrics for state $(\alpha S+\alpha^{-1})$ and $(\alpha^{-1}S+s_2^b)$ are calculating according

$$M_{\alpha^{-1}S+\alpha^{-1}}(k+1) = M_{\alpha^{-1}S+s_2^b}(k+1) = z_k \min[M_S(k)+1,\ M_{S+(s_2^b+\alpha^{-1})\alpha}(k)+1]+$$

$$+\ \bar{z}_k \min[M_{S+\alpha^{-\nu}}(k)+1,\ M_{S+(s_2^b+\alpha^{-1})\alpha+\alpha^{-\nu}}(k)+1] \quad . \quad (7)$$

The states with a term (α^{-1}) are called odd states. The others are called even states. For the even states $(\alpha^{-1}S)$ and $(\alpha^{-1}S+s_2^b+\alpha^{-1})$, the metrics are calculated according to

$$M_{\alpha^{-1}S}(k+1) = z_k \min[M_{S+\alpha^{-\nu}}(k),\ M_{S+\alpha^{-\nu}+(s_2^b+\alpha^{-1})\alpha}(k)+2]+$$

$$+\ \bar{z}_k \min[M_S(k),\ M_{S+(s_2^b+\alpha^{-1})\alpha}(k)+2] \quad , \qquad (8)$$

and

$$M_{\alpha^{-1}S+s_2^b+\alpha^{-1}}(k+1) = z_k \min[M_{S+\alpha^{-\nu}}(k)+2,\ M_{S+\alpha^{-\nu}+(s_2^b+\alpha^{-1})\alpha}(k)] +$$

$$+\ \bar{z}_k \min[M_S(k)+2,\ M_{S+(s_2^b+\alpha^{-1})\alpha}(k)] \quad , \qquad (9)$$

respectively. The hardware sections for the even and odd state metrics are given in *Fig. 8* and *Fig. 9*, respectively.

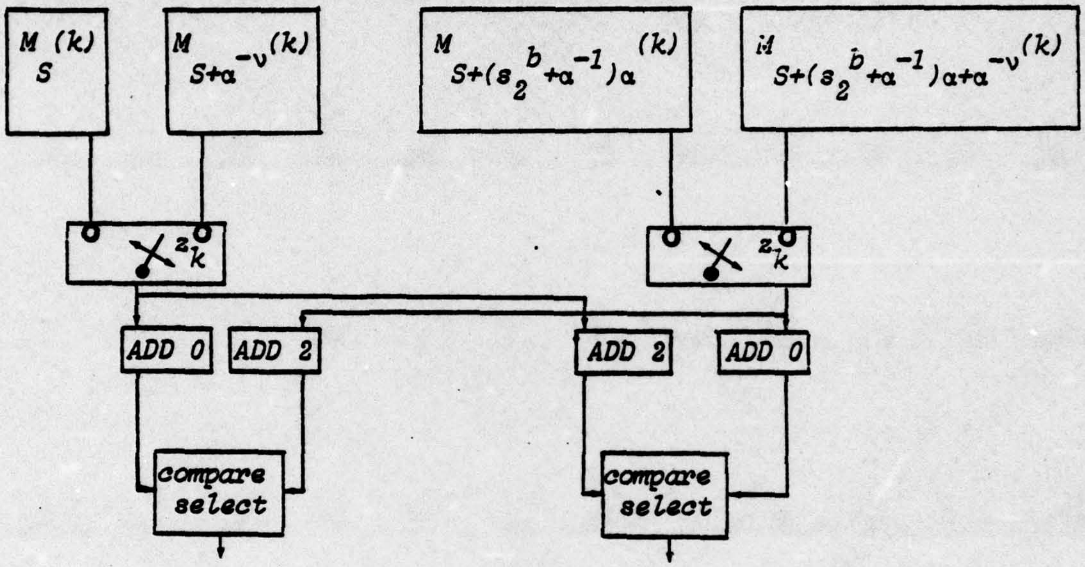
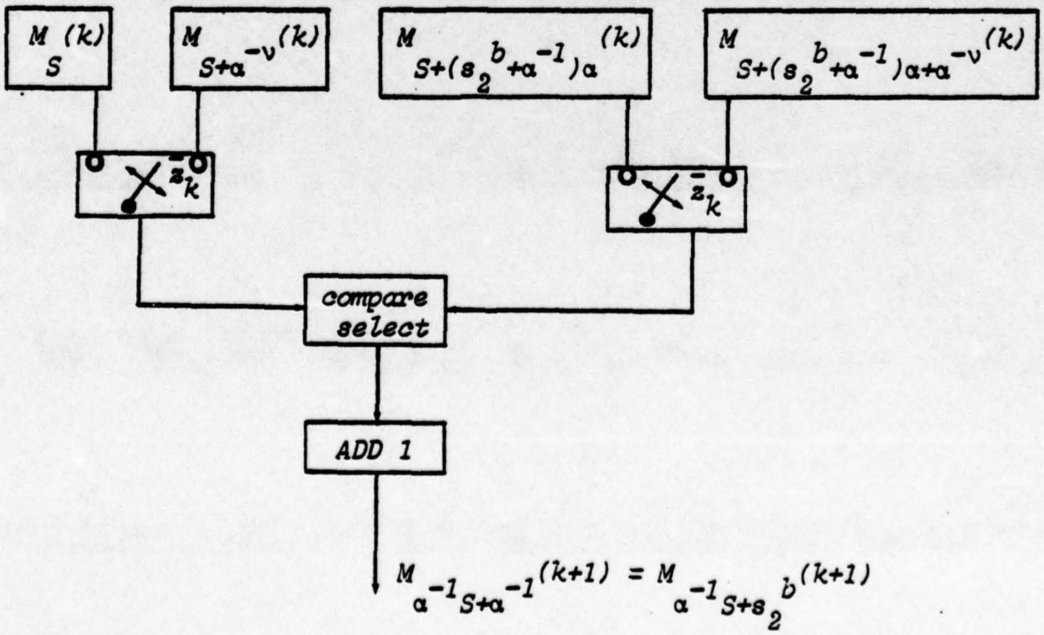


Fig. 8. Metric calculation scheme for an even state pair



$M_S(k)$
$M_{S+\alpha^{-\nu}}(k)$
$M_{S+(s_2^b+\alpha^{-1})\alpha}(k)$
$M_{S+(s_2^b+\alpha^{-1})\alpha+\alpha^{-\nu}}(k)$
$\bar{z}_k$
$\bar{z}_k$
compare select
ADD 1
$M_{\alpha^{-1}S+\alpha^{-1}}(k+1) = M_{\alpha^{-1}S+s_2^b}(k+1)$

Fig. 9 Metric calculation scheme for an odd state pair

The total amount of hardware, needed to calculate the new state metrics follows from *Fig. 6, 8* and *9*, and is given in *Fig. 10*

	adders	*comparators*	*multiplexers*
Syndrome decoder	$2^{\nu-2}*2+2^{\nu-2}$	$2^{\nu-2}*2+2^{\nu-2}$	$2^{\nu-2}*2+2^{\nu-1}$
Viterbi decoder	$2^{\nu-2}*8$	$4.2^{\nu-2}$	-

Fig. 10. Hardware needed to calculate the new state metrics

If equal complexity for all elements is assumed, the syndrome decoder requires $10*2^{\nu-2}$ elements, while the Viterbi decoder does $12*2^{\nu-2}$. In fact, this difference is even larger, because multiplexing is less complex as compared to adding. An additional advantage is the number of different state metrics of a syndrome decoder. In [1] is proven that in general this number is equal to $(\frac{3}{4})*2^{\nu}$. Hence, less subtractions from each state metric with the most likely state metric, and less comparisons for searching this most likely metric need to be made. Next we are discussing the complexity of the path register scheme. The comparator in the metric calculator, determines which path is the survivor for a certain state. This means that in a Viterbi decoder, each path register has two possible entries, as can be seen in *Fig. 11*

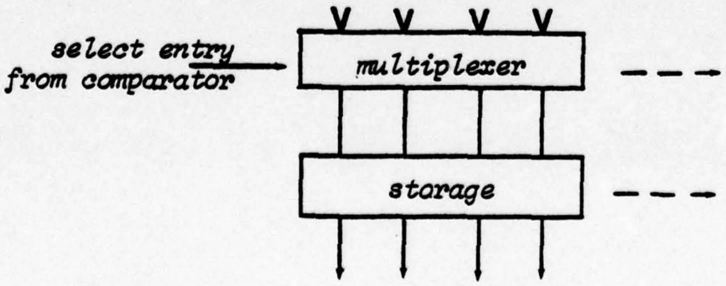


Fig. 11. Four entries of a path register

In general, the path registers of the syndrome decoder, have 4 entries, as can be seen in *Fig. 11*. The multiplexing for two even states is

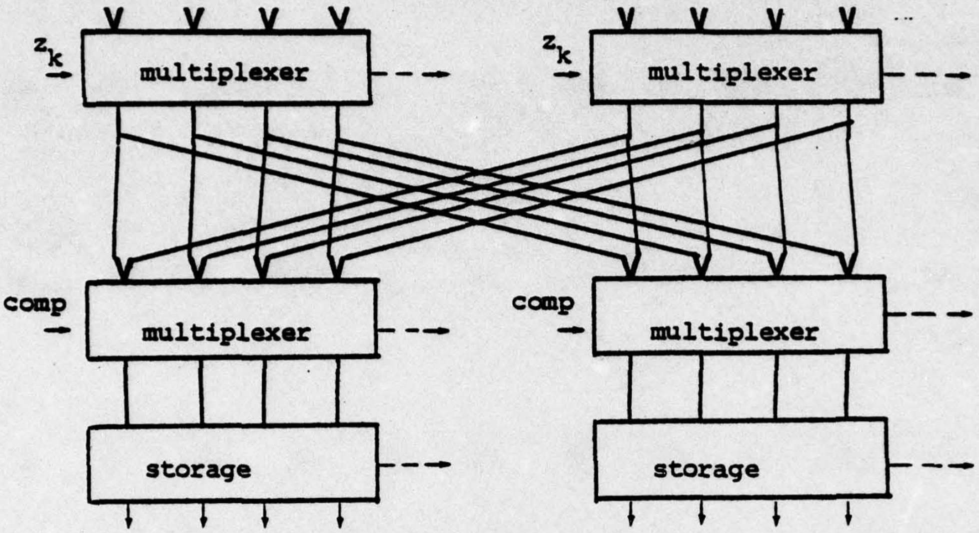


Fig. 12. Path register organization scheme for an even pair

combined, as was done in *Fig. 8* for the metric calculation. The same scheme can be drawn for odd pairs of states. If a path register length of 4 is assumed, the hardware needed for the path register organization in the syndrome decoder and in the Viterbi decoder is given in *Fig. 13*, respectively.

	storage	*multiplexers*
syndrome	$2*2^{\nu-2}*\frac{4\nu}{4}+2^{\nu-2}*\frac{4\nu}{4}$	$4*2^{\nu-2}*\frac{4\nu}{4}+2^{\nu-2}*\frac{4\nu}{4}*3$
Viterbi	$\frac{4\nu}{4}*2^{\nu}.$	$\frac{4\nu}{4}*2^{\nu}$

Fig. 13. Hardware requirement for the path register organization scheme

If equal complexity is assumed for multiplexers and storage cells, the difference between syndrome and Viterbi path register organization is $2*\nu*2^{\nu-2}$ elements in favor of the Viterbi decoder. Together with the metric calculation scheme, we can say that the complexity of the syndrome decoder and Viterbi decoder is about equal.

But as proven in [1], the number of different states in the syndrome decoder can be reduced to $(\sqrt{3})^{\nu}$. Hence, for this special class of codes, the syndrome decoder is a reasonable alternative to the Viterbi decoder, as it achieves an exponentional saving in hardware. Note, that this is only true for $Q=2$, i.e. hard decisions. If soft decisions are used, the number of different states cannot be reduced and the Viterbi decoder is preferable.

REFERENCES

[1] J.P.M. Schalkwijk, "Symmetries of the State Diagram of the Syndrome Former of a Binary Rate-½ Convolutional Code", Proceedings of the C.I.S.M. advanced School on Open Problems in Information Theory, September 1975.

[2] J.A. Heller and I.M. Jacobs, "Viterbi Decoding for Satellite and Space Communication", IEEE Trans. on Comm. Techn., Vol. COM-18, October 1971, pp. 835-848.

4.0 UNION BOUND ON THE ERROR PROBABILITY

EXPLICIT EVALUATION OF VITERBI'S UNION BOUNDS FOR THE FIRST EVENT ERROR PROBABILITY AND THE BIT ERROR PROBABILITY OF A BINARY CONVOLUTIONAL CODE ON A BINARY SYMMETRIC CHANNEL

by

K.A. Post

Abstract. An explicit method is given to evaluate Viterbi's union bounds [1] on both the first event error probability and the bit error probability of binary convolutional codes on a BSC. These bounds are explicitly given for the rate ½ code with generators $1 + D + D^2$ and $1 + D^2$. Comparison is made with bounds and experimental results of Van de Meeberg [2].

1. Let p and q be positive numbers, $p < \frac{1}{2}$, $p + q = 1$.
Following Viterbi [1] we define a sequence

$$P_1, P_2, P_3, \ldots$$

by the formulas

$$(1) \qquad \begin{cases} P_{2k-1} := \sum\limits_{j=0}^{k-1} \binom{2k-1}{j} p^{2k-1-j} q^j \\ P_{2k} := \sum\limits_{j=0}^{k-1} \binom{2k}{j} p^{2k-j} q^j + \frac{1}{2}\binom{2k}{k} p^k q^k \end{cases} \qquad (k = 1,2,3,\ldots).$$

Using the well-known addition property in Pascal's triangle we find

$$(p + q)P_{2k-1} = P_{2k}, \quad \text{so that } P_{2k} = P_{2k-1} \qquad (k = 1,2,3,\ldots)$$

and furthermore

$$(p + q)[P_{2k} - \tfrac{1}{2}\binom{2k}{k} p^k q^k] = P_{2k+1} - p\binom{2k}{k} p^k q^k,$$

so that

$$P_{2k+1} = P_{2k} - (\tfrac{1}{2} - p)\binom{2k}{k} p^k q^k \qquad (k = 1,2,3,\ldots).$$

Combining these results we obtain

$$P_1 = P_2 = \tfrac{1}{2} - (\tfrac{1}{2} - p)\binom{0}{0}$$

$$P_3 = P_4 = \tfrac{1}{2} - (\tfrac{1}{2} - p)[\binom{0}{0} + \binom{2}{1}pq]$$

$$P_5 = P_6 = \tfrac{1}{2} - (\tfrac{1}{2} - p)[\binom{0}{0} + \binom{2}{1}pq + \binom{4}{2}p^2q^2]$$

$$P_7 = P_8 = \tfrac{1}{2} - (\tfrac{1}{2} - p)[\binom{0}{0} + \binom{2}{1}pq + \binom{4}{2}p^2q^2 + \binom{6}{3}p^3q^3]$$

....

It is a well-known fact that for complex z, $|z| < \frac{1}{4}$

(2) $$(1 - 4z)^{-\frac{1}{2}} = \binom{0}{0} + \binom{2}{1}z + \binom{4}{2}z^2 + \binom{6}{3}z^3 + \ldots$$

so that the generating function F for the sequence $P_1, P_2, P_3, \ldots$ must read as follows

(3) $$F(z) := \sum_{k=1}^{\infty} P_k z^k = \frac{z}{1 - z}[\tfrac{1}{2} - (\tfrac{1}{2} - p)(1 - 4pqz^2)^{-\frac{1}{2}}] .$$

The Taylor-series in this formula converges for complex z, $|z| < (4pq)^{-\frac{1}{2}}$, since $z = 1$ is a removable singularity of the function F.

2. Let G be a *rational* function of the complex variable z, $G(z) = \frac{n(z)}{d(z)}$, where n and d are polynomials having gcd one, and $n(0) = 0$. G has a Taylor-expansion around $z = 0$;

(4) $$G(z) = \sum_{k=1}^{\infty} g_k z^k$$

For our purpose we are interested in finding an explicit form for the expression

$$\sum_{k=1}^{\infty} P_k g_k .$$

Recall that the polynomial d can be uniquely factorized

$$(5) \qquad d(z) = C(z - \alpha_1)^{m_1}(z - \alpha_2)^{m_2} \dots (z - \alpha_r)^{m_r} ,$$

where C is a complex constant and α_j are the distinct zeros of d(z) with multiplicities m_j $(j = 1,\dots,r)$.
Then G(z) can be decomposed uniquely into partial fractions

$$(6) \qquad G(z) = q(z) + \sum_{j=1}^{r} \sum_{\ell=1}^{m_j} \frac{A_{j\ell}}{(1 - \frac{z}{\alpha_j})^{\ell}} ,$$

where q is a polynomial and $A_{j\ell}$ are complex constants, $q(0) + \sum_{j,\ell} A_{j\ell} = 0$. Hence we obtain

$$(7) \qquad g_k = q_k + \sum_{j,\ell} A_{j\ell}\, h_{j\ell k} \qquad (k = 1,2,3,\dots),$$

where q_k is the k-th term of q(z) and $h_{j\ell k}$ is the k-th Taylor-coefficient of $(1 - \frac{z}{\alpha_j})^{-\ell}$.

Let us consider this Taylor-coefficient separately. By the binomial series-expansion we have

$$(8) \qquad (1 - \frac{z}{\alpha})^{-\ell} = \sum_{k=0}^{\infty} \binom{\ell+k-1}{\ell-1} \frac{z^k}{\alpha^k} ,$$

and hence, by repeated application of the addition property in Pascal's triangle

$$(9) \qquad (1 - \frac{z}{\alpha})^{-\ell} = \sum_{k=0}^{\infty} \sum_{t=0}^{\ell-1} \binom{k}{t}\binom{\ell-1}{t} \frac{z^k}{\alpha^k} .$$

Therefore, the k-th Taylor-coefficient of $(1 - \frac{z}{\alpha_j})^{-\ell}$ is equal to

$$(10) \qquad h_{j\ell k} = \sum_{t=0}^{\ell-1} \binom{\ell-1}{t}\binom{k}{t} \alpha_j^{-k} .$$

Recall that $\binom{k}{t}\alpha_j^{-k+t} = \frac{1}{t!}(z^k)^{(t)}_{z=\frac{1}{\alpha_j}}$, where $(z^k)^{(t)}_{z=\frac{1}{\alpha_j}}$ denotes the value of the t-th derivative of the function z^k taken at $z = \frac{1}{\alpha_j}$. Hence, the contribution of the term $A_{j\ell}\,(1-\frac{z}{\alpha_j})^{-\ell}$ to the value of the desired expression $\sum_{k=1}^{\infty} g_k P_k$ is equal to

$$A_{j\ell}\sum_{t=0}^{\ell-1}\binom{\ell-1}{t}\frac{1}{t!\alpha_j^{t}}\,F^{(t)}\left(\frac{1}{\alpha_j}\right) ,$$

provided that $\left|\frac{1}{\alpha_j}\right| < (4pq)^{-\frac{1}{2}}$ (cf. (3)).

3. We now apply our results to the well-known convolutional code over GF(2) with generator polynomials $1 + D + D^2$ and $1 + D^2$.
The function $G = G_E$ to be taken for the construction of the union bound for the first event error probability P_E in [1] is equal to

$$G_E(z) := \frac{z^5}{1-2z} = -\frac{1}{32} - \frac{1}{16}z - \frac{1}{8}z^2 - \frac{1}{4}z^3 - \frac{1}{2}z^4 + \frac{1}{32}\,\frac{1}{1-2z} .$$

So we find for the union bound for P_E

$$P_E < -\frac{1}{16}P_1 - \frac{1}{8}P_2 - \frac{1}{4}P_3 - \frac{1}{2}P_4 + \frac{1}{32}F(2)$$

$$= \frac{1}{32}(1-2p)(1-16pq)^{-\frac{1}{2}} - \frac{1}{32} - \frac{3}{16}p - \frac{9}{4}p^2 + \frac{3}{2}p^3 .$$

For the construction of the union bound for the bit error probability P_B in [1] we must take $G = G_B$:

$$G_B(z) := \frac{z^5}{(1-2z)^2} = \frac{1}{8} + \frac{3}{16}z + \frac{1}{4}z^2 + \frac{1}{4}z^3 - \frac{5}{32}\,\frac{1}{1-2z} + \frac{1}{32}\,\frac{1}{(1-2z)^2} .$$

Hence we find as union bound for P_B

$$P_B < \frac{3}{16}P_1 + \frac{1}{4}P_2 + \frac{1}{4}P_3 - \frac{5}{32}F(2) + \frac{1}{32}[F(2) + 2F'(2)]$$

$$= \frac{5}{32} + \frac{7}{16}p + \frac{3}{4}p^2 - \frac{1}{2}p^3 - \frac{1}{32}(1-2p)(5-96pq)(1-16pq)^{-\frac{3}{2}} .$$

These bounds hold for $pq < \frac{1}{16}$.

4. The first attempt to find an upper bound for the union bounds for P_E and P_B was made by Viterbi [1]. Van de Meeberg [2] improved this upper bound using only the fact that $P_{2k} = P_{2k-1}$ $(k = 1,2,3,\ldots)$. Experimental measurements also have been made on P_B for the special code with generators $1 + D + D^2$ and $1 + D^2$. It turns out numerically that the union bound on P_B for this code is closer to Van de Meeberg's bound than to the experimental results. So it seems to be useful now to study the philosophy behind Viterbi's union bound in order to obtain better bounds.

[1] A.J. Viterbi. Convolutional codes and their performance in communications systems.
IEEE Trans. Comm. Techn. COM-19 (1971), 751-772.

[2] L. van de Meeberg. A tightened upper bound on the error probability of binary convolutional codes with Viterbi decoding.
IEEE Trans. Inf. Theory IT-20 (1974), 389-391.

5.0 CONCLUSIONS

Our main result, as described in Section 2.0, is a state space formalism that can be used to advantage in formulating and solving certain problems in the theory of convolutional codes. We were able to identify certain state space symmetries that allow for an exponential reduction of the hardware required for the implementation of a maximum likelihood decoder. However, the possibilities of our state space formalism are by no means exhausted. For example, using this formalism 1) we are able to improve on existing bounds on the free distance of fixed convolutional codes, 2) we are investigating the possibilities of hardware reductions in maximum likelihood decoders for ternary convolutional codes, 3) we are investigating possible savings in the amount of computation in sequential decoding, 4) we are searching for other symmetries that could possibly lead to even larger hardware savings, etc. In conclusion, our present research effort concerns the relation between performance and complexity in convolutional coding, where these problems are formulated in a state space framework.

APPENDIX

SYNDROME DECODING OF BINARY RATE 1/2 CONVOLUTIONAL CODES

Syndrome Decoding of Binary Rate-½ Convolutional Codes

J. PIETER M. SCHALKWIJK, MEMBER, IEEE, AND A. J. VINCK

Abstract—The classical Viterbi decoder recursively finds the trellis path (code word) closest to the received data. Given the received data, the syndrome decoder first forms a syndrome instead. Having found the syndrome, that only depends on the channel noise, a recursive algorithm like Viterbi's determines the noise sequence of minimum Hamming weight that can be a possible cause of this syndrome. Given the estimate of the noise sequence, one derives an estimate of the original data sequence. The bit error probability of the syndrome decoder is no different from that of the classical Viterbi decoder. However, for short constraint length codes the syndrome decoder can be implemented using a read-only memory (ROM), thus obtaining a considerable saving in hardware. The syndrome decoder has at most $\frac{3}{4}$ as many path registers as does the Viterbi decoder. There exist convolutional codes for which the number of path registers can be even further reduced.

Paper approved by the Editor for European Contributions of the IEEE Communications Society for publication after presentation at the Information Theory Colloquium, Ketzhely, Hungary. Manuscript received September 29, 1975; revised April 15, 1976. This research was supported in part by the U.S. Navy under Contract N00123-75-C-0557.

The authors are with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.

I. INTRODUCTION

THIS paper extends and generalizes some earlier results [1] on syndrome decoding of rate-$\frac{1}{2}$ convolutional codes. The binary code generated by the encoder of Fig. 1 will again be used as an example throughout the paper. The additions in Fig. 1 are modulo-2 and all binary sequences $\cdots, b_{-1}, b_0, b_1, \cdots$ are represented as power series $b(\alpha) = \cdots + b_{-1}\alpha^{-1} + b_0 + b_1\alpha + \cdots$. The encoder has connection polynomials $C_1(\alpha) = 1 + \alpha^2$ and $C_2(\alpha) = 1 + \alpha + \alpha^2$. In general, the encoder outputs are $C_1(\alpha)x(\alpha)$ and $C_2(\alpha)x(\alpha)$. The syndrome $z(\alpha)$ only depends on $n_1(\alpha)$ and $n_2(\alpha)$, i.e., not on the data sequence $x(\alpha)$, for

$$\begin{aligned} z(\alpha) &= C_2(\alpha)[C_1(\alpha)x(\alpha) + n_1(\alpha)] + C_1(\alpha)[C_2(\alpha)x(\alpha) + n_2(\alpha)] \\ &= C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha). \end{aligned} \tag{1}$$

Having formed the syndrome $z(\alpha)$, Section III describes a recursive algorithm like Viterbi's [2] to determine from the

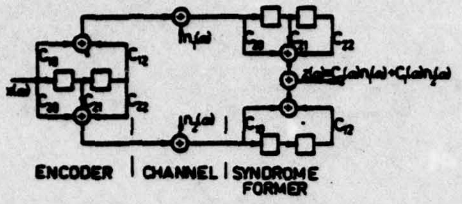


Fig. 1. Encoding and syndrome forming for a $R = \frac{1}{2}$ code.

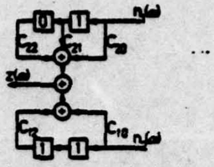


Fig. 2. Syndrome former in its base state.

syndrome $z(\alpha)$ the noise sequence pair $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ of minimum Hamming weight that can be a possible cause of this syndrome.

Given the estimate $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ of the noise sequence pair, one derives an estimate $\hat{x}(\alpha)$ of the original data sequence $x(\alpha)$ as follows. For a noncatastrophic code, $C_1(\alpha)$ and $C_2(\alpha)$ are relatively prime. Hence, by Euclid's algorithm [3] there exist polynomials $D_1(\alpha)$ and $D_2(\alpha)$ such that $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$. For the example of Fig. 1, we have $D_1(\alpha) = 1 + \alpha$, $D_2(\alpha) = \alpha$. We receive the sequence pair

$$y_i(\alpha) = C_i(\alpha)x(\alpha) + n_i(\alpha), \qquad i = 1, 2, \tag{2}$$

and form the estimate

$$\hat{x}(\alpha) = D_1(\alpha)[y_1(\alpha) + \hat{n}_1(\alpha)] + D_2(\alpha)[y_2(\alpha) + \hat{n}_2(\alpha)]. \tag{3}$$

Note that if the noise sequence estimate $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ is correct we have

$$\begin{aligned} y_i(\alpha) + \hat{n}_i(\alpha) &= C_i(\alpha)x(\alpha) + n_i(\alpha) + \hat{n}_i(\alpha) \\ &= C_i(\alpha)x(\alpha), \qquad i = 1, 2, \end{aligned}$$

and, hence

$$\hat{x}(\alpha) = D_1(\alpha)C_1(\alpha)x(\alpha) + D_2(\alpha)C_2(\alpha)x(\alpha) = x(\alpha).$$

Note that (3) for the estimate $\hat{x}(\alpha)$ of the data sequence $x(\alpha)$ can be rewritten as

$$\hat{x}(\alpha) = [D_1(\alpha)y_1(\alpha) + D_2(\alpha)y_2(\alpha)] + \hat{\omega}(\alpha), \tag{4}$$

where

$$\omega(\alpha) = D_1(\alpha)n_1(\alpha) + D_2(\alpha)n_2(\alpha), \tag{5}$$

and $\hat{\omega}(\alpha)$ equals the right-hand side (RHS) of (5) with $\hat{n}_i(\alpha)$ substituted for $n_i(\alpha)$, $i = 1, 2$. The term in square brackets in (4) can be computed directly from the received data using very simple circuitry. As there is no need to distinguish between pairs $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ and $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]'$ that lead to the same value for $\hat{\omega}(\alpha)$ in (4), the algorithm to be discussed in Section III computes $\hat{\omega}(\alpha)$ directly.

II. STATE DIAGRAM

In Fig. 2 we have redrawn the syndrome former of our example. As, according to (1), the syndrome $z(\alpha)$ only depends on the noise pair $[n_1(\alpha),n_2(\alpha)]$, all other binary sequences have been omitted from Fig. 2. For minimum distance decoding we are now presented with the following problem. Given the syndrome $z(\alpha)$, determine the noise pair $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ of minimum Hamming weight that can be a cause of this syndrome. Before tackling this problem in Section III, it will be necessary to first derive some general properties of the state diagram of a syndrome former for a binary rate -$\frac{1}{2}$ convolutional code.

Let ν be the number of memory stages of the encoder, i.e., $\nu = 2$ for the encoder of Fig. 1. The corresponding syndrome former of Fig. 2 has $2^{2\nu} = 16$ "physical states," where a physical state is defined as the contents $S = [s_1(\alpha),s_2(\alpha)]$, where

$$s_i(\alpha) = s_{i,-\nu}\alpha^{-\nu} + s_{i,-\nu+1}\alpha^{-\nu+1} + \cdots + s_{i,-1}\alpha^{-1}, \qquad i = 1, 2,$$

of the $2\nu = 4$ memory cells of the syndrome former. Thus, at first sight, the state diagram of the syndrome former appears more complicated than the state diagram used to implement the classical Viterbi decoder [2], that has only $2^\nu = 4$ states. However, on closer inspection, it turns out that the $2^{2\nu} = 16$ physical states of the syndrome former can be divided into $2^\nu = 4$ equivalence classes or "abstract states," where any two physical states in the same equivalence class give the same output $z(\alpha)$ irrespective of the input pair $[n_1(\alpha),n_2(\alpha)]$. In general, to prove the existence of the 2^ν, $\nu = 1, 2, \cdots$, abstract states defined above, we need the following definitions. As the syndrome former is a time invariant circuit we assume without loss of generality that the state S is present at time $t = 0$.

Definition 1: A "zero-equivalent" state is a physical state with the property that if the syndrome former is in such a state, an all-zero input $[n_1(\alpha),n_2(\alpha)]_0^\infty$ gives rise to an all-zero output $[z(\alpha)]_0^\infty$, where $[b(\alpha)]_{k_1}^{k_2}$ indicates that part of the power series $b(\alpha)$ for which $k_1 < \exp \alpha < k_2$.

Definition 2: A "base" state $S^b = [\alpha^{-1},s_2{}^b(\alpha)]$ is a zero-equivalent state with a single "1" in the rightmost position of the top register of the syndrome former, see Fig. 2.

A base state can be constructed as follows. Start with the top and bottom registers of the syndrome former, Fig. 2, all zero. Put a 1 in the leftmost position of the top register. Assuming that $C_1(\alpha)$ and $C_2(\alpha)$ both have a nonzero term of degree ν, we now have to put a 1 in the leftmost position of the bottom register, as otherwise the corresponding digit of the syndrome $z(\alpha)$ would differ from zero. Subsequently, shift the contents of the top and the bottom register one place to the right, feeding 0's into both leftmost positions, respectively. If the corresponding digit of the syndrome $z(\alpha)$ differs from zero set the leftmost position of the bottom register equal to 1, thus complementing the corresponding syndrome digit, etc. This process continues until the single 1 in the top register is in the rightmost position. The bottom register now contains

$s_2{}^b(\alpha)$. It is clear that the above construction leads to a unique result. This base state S^b is indicated in the example of Fig. 2. However, there might conceivably be another construction leading to a different base state. The following theorem shows that this is not the case.

Theorem 1: The base state $S^b = [\alpha^{-1}, s_2{}^b(\alpha)]$ is unique.

Proof: Suppose there are two base states $S_1{}^b$ and $S_2{}^b$. These base states are zero-equivalent states, hence, so is their sum $S = S_1{}^b + S_2{}^b$. But as the sum of two base states the physical state S has all zeros in the top register of the syndrome former

As $C_1(\alpha)$ has a nonzero term of degree ν, the only zero-equivalent state with the top register contents all zero is the all-zero state. Hence, S is the all-zero state and the physical states $S_1{}^b$ and $S_2{}^b$ are equal. Q.E.D.

We will now show that there are 2^ν equivalence classes of physical states and that each class has a unique representative physical state for which the contents of the top register of the syndrome former, Fig. 2, is all zero. It is these representative states $S = [0, s_2(\alpha)]$, to be referred to as "the states," that will be used in the remainder of the paper.

Theorem 2: The $2^{2\nu}$, $\nu = 1, 2, \cdots$, physical states of the syndrome former, corresponding to a binary rate-$\frac{1}{2}$ encoder with ν memory cells, can be divided into 2^ν equivalence classes or abstract states. Each equivalence class has a unique representative physical state $S = [0, s_2(\alpha)]$ for which the top register, see Fig. 2, of the syndrome former is all zero.

Proof: Two physical states were related if they resulted in the same output $z(\alpha)$ irrespective of the input pair $[n_1(\alpha), n_2(\alpha)]$. To prove that this relation is an equivalence relation, we must show that it is reflexive, symmetric, and transitive. Reflexivity and symmetry are obvious. To show transitivity let S_1 be related to S_2 and S_2 be related to S_3. Since S_1 and S_2 produce the same output $z(\alpha)$, their sum $S_1 + S_2$ is a zero-equivalent state, as is $S_2 + S_3$. But $S_1 + S_3 = (S_1 + S_2) + (S_2 + S_3)$. Hence, $S_1 + S_3$ is the sum of two zero-equivalent states and thus also zero equivalent. In other words, the physical states S_1 and $S_1 + (S_1 + S_3) = S_3$ produce the same output $z(\alpha)$ and thus the relation defined above is transitive. This completes the first part of the proof. The relation defined above is an equivalence relation and, hence, divides the set of physical states into equivalence classes or abstract states. As the sum of zero-equivalent states is again zero equivalent, left shifts of the base state S^b can be added to the all zero state to obtain a zero-equivalent state for which the top register has any desired contents. Two zero-equivalent states S_1 and S_2 that have the same top register contents are identical, for their sum $S = S_1 + S_2$ is a zero-equivalent state with top register contents all zero. Hence, as shown in the proof of Theorem 1, S is the all-zero state and thus $S_1 = S_2$. In other words, there are 2^ν zero-equivalent states and, in fact, all equivalence classes have 2^ν members, giving $2^{2\nu}/2^\nu = 2^\nu$ abstract states. As we can add left shifts of the base state S^b to any particular physical state without leaving its equivalence class, each equivalence class has a representative member, called the state $S = [0, s_2(\alpha)]$, that has the contents of the top register, Fig. 2, of the syndrome former all zero. To prove uniqueness of the representative state within an equivalence class, assume two representative states S_1 and S_2. The sum $S_1 + S_2$ of two representative states S_1 and S_2 within the same equivalence class is a zero-equivalent state with top register contents all zero. This sum state again must be the all-zero state. Hence, $S_1 + S_2 = 0$ or $S_1 = S_2$, proving that the representative state of an equivalence class is unique. Q.E.D.

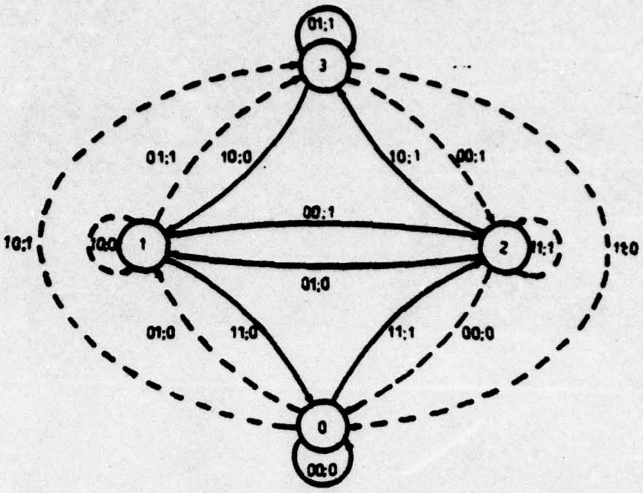


Fig. 3. State diagram of syndrome former.

We are now ready, as an example, to construct the state diagram, see Fig. 3, of the syndrome former of Fig. 2. The states $S_0 = [0,0]$, $S_1 = [0,\alpha^{-1}]$, $S_2 = [0,\alpha^{-2}]$, and $S_3 = [0,\alpha^{-2} + \alpha^{-1}]$ are representative states with the contents of the bottom register, Fig. 2, of the syndrome former equal to 00, 01, 10, and 11, respectively. An input $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [0,1]$ brings us from state S_0 to state S_1. An input $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [1,0]$ brings us from state S_0 to state $S' = [\alpha^{-1},0]$, which is not a representative physical state. The representative state in the equivalence class of S' can be found through addition of the base state $S^b = [\alpha^{-1}, s_2{}^b(\alpha)]$, where $s_2{}^b(\alpha) = \alpha^{-2} + \alpha^{-1}$ from Fig. 2. Hence, $S' + S^b = [0,\alpha^{-2} + \alpha^{-1}]$, i.e., an input $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [1,0]$ brings us from state S_0 to state $S_3 = S' + S^b$. In the same fashion $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [1,1]$ brings us from state S_0 to state S_2. All states in the state diagram of Fig. 3 have now been identified and we leave it to the reader to fill in the remaining edges. A solid edge in Fig. 3 indicates that the syndrome digit corresponding to the particular transition is 0, a dashed edge corresponds to a syndrome digit 1. The numbers next to the edges are the values n_1, n_2; ω. As has been explained in Section I, it is the coefficients $\cdots$, ω_{-1}, ω_0, ω_1, $\cdots$ of the power series $\omega(\alpha)$ of (5) that one is really interested in. It requires some explanation that the generic value ω of these coefficients can also be indicated next to the edges in Fig. 3.

Theorem 3: The value of the coefficient ω_k of the power series $\omega(\alpha) = \cdots + \omega_{-1}\alpha^{-1} + \omega_0 + \omega_1\alpha + \cdots$ defined by (5) is uniquely determined by the state $S(k)$ of the syndrome former at time $t = k$ and by the value of its input $[n_{1k}, n_{2k}]$, $k = \cdots, -1, 0, +1, \cdots$.

Proof: As the syndrome former is a time-invariant circuit, all one must prove is that ω_0 is uniquely determined by $S(0)$ and $[n_{10}, n_{20}]$. Equation (5) can be rewritten as

$$\omega_0 = [D_1'(\alpha)n_1(\alpha) + D_2'(\alpha)n_2(\alpha)]_0{}^0 + (D_{10}n_{10} + D_{20}n_{20}), \tag{6}$$

where the prime on $D_i'(\alpha)$, $i = 1, 2$, indicates that a possible constant term D_{i0} has been omitted. The second term $(D_{10}n_{10} + D_{20}n_{20})$ on the RHS of (6) is completely determined by the input $[n_{10}, n_{20}]$ to the syndrome former at time $t = 0$. Now if the state $S(0)$ of the syndrome former at time $t = 0$ determines $[n_1(\alpha), n_2(\alpha)]_{-\nu}{}^{-1}$ and if deg $D_1'(\alpha) \leqslant \nu$ and deg $D_2'(\alpha) \leqslant \nu$, then $S(0)$ determines the value of the first term $[D_1'(\alpha)n_1(\alpha) + D_2'(\alpha)n_2(\alpha)]_0{}^0$ on the RHS of (6) and we are done. Now from Berlekamp [3, p. 27] we know that there exist polynomials $D_1(\alpha)$ and $D_2(\alpha)$ satisfying $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$ and thus that deg $D_1(\alpha) <$ deg $C_2(\alpha)$ and deg $D_2(\alpha) <$ deg $C_1(\alpha)$. Hence, the degrees of both $D_1'(\alpha)$ and $D_2'(\alpha)$ are less than the common degree ν of $C_1(\alpha)$ and $C_2(\alpha)$. However, the state $S(0)$ determines $[n_1(\alpha), n_2(\alpha)]_{-\nu}{}^{-1}$ only to within an equivalence class. It thus remains to be shown that addition of a zero-equivalent state $S = [s_1(\alpha), s_2(\alpha)]$ does not affect the value of $[D_1'(\alpha)n_1(\alpha) + D_2'(\alpha)n_2(\alpha)]_0{}^0$. For a zero-equivalent state we have by definition

$$[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)]_0{}^\infty = 0. \tag{7}$$

From $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$ it follows that $D_1'(\alpha)C_1(\alpha) + D_2'(\alpha)C_2(\alpha) = 0$. Thus we can define the polynomial $P(\alpha)$ by

$$P(\alpha) = D_1'(\alpha)C_1(\alpha) = D_2'(\alpha)C_2(\alpha). \tag{8}$$

It now follows that

$$\begin{aligned} P(\alpha)&[D_1'(\alpha)s_1(\alpha) + D_2'(\alpha)s_2(\alpha)] \\ &= D_1'(\alpha)D_2'(\alpha)[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)]. \end{aligned} \tag{9}$$

As deg $C_1(\alpha) >$ deg $D_2'(\alpha)$, we have deg $P(\alpha) >$ deg $D_1'(\alpha)D_2'(\alpha)$. Thus, if $[D_1'(\alpha)s_1(\alpha) + D_2'(\alpha)s_2(\alpha)]$ had a nonzero term of degree 0, then $[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)]$ would have a nonzero term of degree larger than 0. As, according to (7), this cannot be the case it follows that $[D_1'(\alpha)s_1(\alpha) + D_2'(\alpha)s_2(\alpha)]_0{}^0 = 0$. Q.E.D.

In fact, we even have the stronger result that all edges leading to the same state $S(k + 1)$, $k = \cdots, -1, 0, +1, \cdots$, have the same value of ω_k associated with them.

Corollary: The value of ω_k, $k = \cdots, -1, 0, +1, \cdots$, is uniquely determined by the value of $S(k + 1)$.

Proof: As the syndrome former is a time-invariant circuit, all one must prove is that ω_0 is uniquely determined by the state $S(1)$ of the syndrome former at time $t = 1$. According to (6), the value of ω_0 is uniquely determined by $[n_1(\alpha), n_2(\alpha)]_{-\nu+1}{}^0$ as the degree of both $D_1(\alpha)$ and $D_2(\alpha)$ is less than the common degree ν of $C_1(\alpha)$ and $C_2(\alpha)$. In fact, we only need to know $[n_1(\alpha), n_2(\alpha)]_{-\nu+1}{}^0$ to within an equivalence class. The proof is identical to the proof of Theorem 3, except that we redefine $P(\alpha)$ of (8), as

$$Q(\alpha) = D_1(\alpha)C_1'(\alpha) + D_2(\alpha)C_2'(\alpha). \tag{10}$$

As $S(1)$, in fact, defines $[n_1(\alpha), n_2(\alpha)]_{-\nu+1}{}^0$ to within such an equivalence class, $S(1)$ uniquely determines ω_0. Q.E.D.

III. ALGORITHM

As the recursive algorithm to be described in this section is similar to Viterbi's [2], we can be very brief. For reasons of clarity the decoding algorithm will be explained using the code generated by the encoder of Fig. 1 as an example. Fig. 4 represents the kth section, $k = \cdots, -1, 0, +1, \cdots$, of the trellis diagram corresponding to the state diagram of Fig. 3. The decoding algorithm is to find the coefficients ω_k of the power series $\omega(\alpha) = \cdots + \omega_{-1}\alpha^{-1} + \omega_0 + \omega_1\alpha + \cdots$ associated with the path of minimum weight through the trellis diagram. The pertinent weight is the Hamming weight of the pair $[\hat{n}_1(\alpha), \hat{n}_2(\alpha)]$ associated with the particular path. As in the Viterbi algorithm [2], to find the minimum weight path we associate a metric with each possible state. The metrics at time $t = k$ can be computed recursively given the metrics at time $t = k - 1$. For the trellis diagram of Fig. 4, the recursion is given by

$$\begin{aligned} M_0(k+1) = \bar{z}_k \min\,&[M_0(k), M_1(k) + 2] \\ &+ z_k \min\,[M_2(k), M_3(k) + 2] \end{aligned} \tag{11a}$$

$$\begin{aligned} M_1(k+1) = \bar{z}_k \min\,&[M_2(k) + 1, M_3(k) + 1] \\ &+ z_k \min\,[M_0(k) + 1, M_1(k) + 1] \end{aligned} \tag{11b}$$

$$\begin{aligned} M_2(k+1) = \bar{z}_k \min\,&[M_0(k) + 2, M_1(k)] \\ &+ z_k \min\,[M_2(k) + 2, M_3(k)] \end{aligned} \tag{11c}$$

$$\begin{aligned} M_3(k+1) = \bar{z}_k \min\,&[M_2(k) + 1, M_3(k) + 1] \\ &+ z_k \min\,[M_0(k) + 1, M_1(k) + 1], \end{aligned} \tag{11d}$$

where $\bar{z}_k$ is the modulo-2 complement of z_k, $k = \cdots, -1, 0, +1, \cdots$. Note that for each value $z_k = 0$ or $z_k = 1$ two arrows impinge on each $(k + 1)$-state. The arrow associated with the minimum within the relevant pair of square brackets in (11) is called the "survivor." If both arrows have this property, flip a coin to determine the survivor. In the classical Viterbi [2] implementation of the algorithm each state S_j, $j = 0, 1, 2, 3$, has a metric register MR$_j$ and a path register PR$_j$ associated with it. The metric register is used to store the current metric value $M_j(k + 1)$ at time $t = k$, $k = \cdots, -1, 0, +1, \cdots$, associated with state S_j, $j = 0, 1, 2, 3$. The path register PR$_j$[0 : $D - 1$] stores the ω-values associated with the current sequence of the D most recent survivors leading up to state S_j at time $t = k$. The pertinent output is

$$\hat{\omega}_{k-D} = \text{CONTENTS PR}_{j(k)}[D - 1 : D - 1], \tag{12a}$$

where $j(k)$ minimizes $M_j(k + 1)$, i.e.,

$$M_{j(k)}(k+1) = \min_j M_j(k+1). \tag{12b}$$

If more than one j satisfies (12b), select $j(k)$ arbitrarily among the candidates. As the algorithm returns $\hat{\omega}_{k-D}$ at time $t = k$,

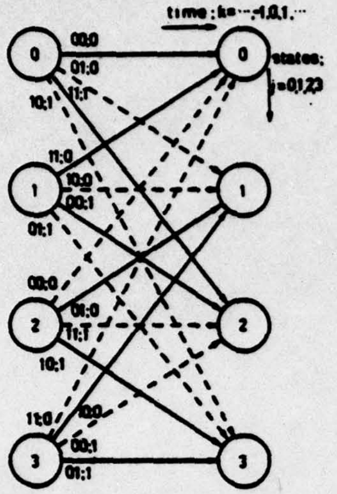


Fig. 4. kth section of the trellis diagram, $k = \cdots, -1, 0, +1, \cdots$.

$k = \cdots, -1, 0, +1, \cdots$, the path register length D is also referred to as the coding delay. The resulting bit error probability P_b decreases as the coding delay D increases. Increasing D beyond $5(\nu + 1)$ does not lead to appreciable further decrease in the value of P_b. This relation between the bit error probability P_b and the coding delay D will be elaborated on further in Section VI. The next section is concerned with a practical implementation of the syndrome decoder.

IV. IMPLEMENTATION

So far, the syndrome decoder has only been of theoretical interest as a possible alternative for the classical Viterbi decoder [2]. We will now study a practical implementation and in the next section make some comparisons as to the relative hardware complexity of these competing decoders.

Using (11) we construct Table I. The first column just numbers the rows of the table. The second column lists all possible metric combinations $M_0(k)$, $M_1(k)$, $M_2(k)$, $M_3(k)$ at time $k-1$. As only the differences between the metrics of a quadruple matter, we substract from each member of a quadruple of metrics the minimum value of the quadruple, i.e., all quadruples of metrics in Table I have one or more zeros. Columns 3 and 4 apply to the case that $z_k = 0$ and columns 5 and 6 to the case that $z_k = 1$. Columns 3 and 5 list the survivors, i.e., the indices of the associated $(k-1)$ states, and columns 4 and 6 the new metrics $M_0(k+1)$, $M_1(k+1)$, $M_2(k+1)$, $M_3(k+1)$ as given by (11). If there is a choice of survivors, the candidates are placed within parentheses in the survivor columns.

Table I contains more information than is necessary for the actual implementation of the syndrome decoder. As explained in Section III, knowledge of the survivor leading to each state, together with the index j_m of the minimum within each new quadruple of metrics, suffices to determine the key sequence $\omega(\alpha)$ of (5). Hence, we omit the quadruples of metrics from Table I and store the resulting Table II in a read-only memory (ROM). The operation of the core part of the syndrome decoder can now be explained using the block diagram of Fig. 5. Assume that at time k the ROM address register AR contains (AR) = 7 and the ROM data register DR contains (DR) = (ROM,7). Note, see Fig. 4 and also the corollary to Theorem 3, that the ω-values to be shifted into $\mathrm{PR}_0[0:0]$, $\mathrm{PR}_1[0:0]$, $\mathrm{PR}_2[0:0]$, $\mathrm{PR}_3[0:0]$ are 0011, respectively. Let $z_k = 1$. Then according to row 7 and column 5 of Table II, or according to the contents (DR) of the DR, replace

$$\mathrm{PR}_0[1:D-1] \leftarrow \text{CONTENTS } \mathrm{PR}_2[1:D-1]$$

$$\mathrm{PR}_1[1:D-1] \leftarrow \text{CONTENTS } \mathrm{PR}_1[1:D-1]$$

$$\mathrm{PR}_2[1:D-1] \leftarrow \text{CONTENTS } \mathrm{PR}_3[1:D-1]$$

$$\mathrm{PR}_3[1:D-1] \leftarrow \text{CONTENTS } \mathrm{PR}_1[1:D-1].$$

The rightmost digit, $\mathrm{PR}_0[D-1:D-1]$, $\mathrm{PR}_1[D-1:D-1]$, $\mathrm{PR}_2[D-1:D-1]$, $\mathrm{PR}_3[D-1:D-1]$, of all four path registers is fed to the selector, see Fig. 5, that determines $\hat{\omega}_{k-D}$ according to (12a) using as $j(k)$ the entry in row 7 and column 7, i.e., $j_m = 2$, of Table II which can also be found in the DR. To complete the kth cycle of the syndrome decoder, set (AR) = 8 and read DR ← (ROM,8). The ROM decoder for the code of Fig. 1 has been realized in hardware using path registers of length $D = 11$. The experimental results will be discussed in Section VI.

TABLE I
METRIC TRANSITIONS

		$z_k = 0$					$z_k = 1$				
Row Number	Old Metrics	Survivors				New Metrics	Survivors				New Metrics
0	0000	0	(2,3)	1	(2,3)	0101	2	(0,1)	3	(0,1)	0101
1	0101	0	2	1	2	0111	2	0	3	0	0111
2	0111	0	(2,3)	1	(2,3)	0212	2	0	3	0	0000
3	0212	0	2	(0,1)	2	0222	2	0	3	0	0010
4	0222	0	(2,3)	(0,1)	(2,3)	0323	2	0	3	0	1010
5	0010	0	3	1	3	0101	2	(0,1)	3	(0,1)	1101
6	0323	0	2	0	2	0323	2	0	3	0	1020
7	1010	0	3	1	3	1101	2	1	3	1	1101
8	1101	0	2	1	2	0000	2	(0,1)	3	(0,1)	0212
9	1020	0	3	1	3	1101	(2,3)	1	3	1	2101
10	2101	0	2	1	2	1000	2	1	3	1	0212
11	1000	0	(2,3)	1	(2,3)	1101	2	1	3	1	0101

V. PATH REGISTER SAVINGS

The ROM-implementation of the syndrome decoder as described in Section IV has been realized for the codes listed in column 2 of Table III. We will discuss some interesting aspects of this table. The first row lists several properties of code 1 that was used as an example throughout the earlier part of the paper. Column 3 lists the number of metric combinations of the various codes. The classical Viterbi decoder [2] can also be realized using a ROM in the manner described in Section IV. However, the Viterbi decoder for code 1 has 31 metric combinations, whereas the syndrome decoder has only 12 metric combinations. For the $\nu = 4$ codes this difference is even more pronounced. For codes 2 and 3 the syndrome decoder has, respectively, 1686 and 1817 metric combinations, whereas the classical Viterbi decoder for either of these codes has more than 15 000 metric combinations. Note that in

TABLE II
CONTENTS OF THE ROM

	$z_k = 0$			$z_k = 1$		
Old ROM Address	Survivors	New ROM Address	Index j_m	Survivors	New ROM Address	Index j_m
0	0(2,3) 1 (2,3)	1	(0,2)	2 (0,1) 3 (0,1)	1	(0,2)
1	0 2 1 2	2	0	2 0 3 0	2	0
2	0(2,3) 1 (2;3)	3	0	2 0 3 0	0	(0,1,2,3)
3	0 2 (0,1? 2	4	0	2 0 3 0	5	(0,1,3)
4	0(2,3) (0,1) (2,3)	6	0	2 0 3 0	7	(1,3)
5	0 3 . 1 3	1	(0,2)	2 (0,1) 3 (0,1)	8	2
6	0 2 0 2	6	0	2 0 3 0	9	(1,3)
7	0 3 1 3	8	2	2 1 3 1	8	2
8	0 2 1 2	0	(0,1,2,3)	2 (0,1) 3 (0,1)	3	0
9	0 3 1 3	8	2	(2,3) 1 3 1	10	2
10	0 2 1 2	11	(1,2,3)	2 1 3 1	3	0
11	0(2,3) 1 (2,3)	8	2	2 1 3 1	1	(0,2)

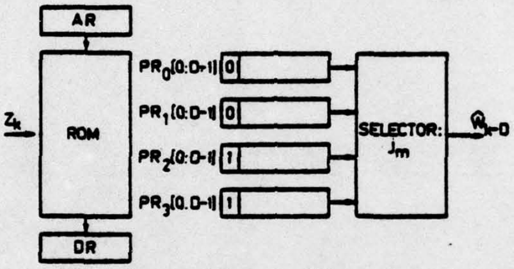


Fig. 5. Block diagram of the core of the syndrome decoder.

TABLE III
CODES REALIZED

code	connection polynomials; 0-th order right	number of metric combinations	minimum number of path registers	distance	number of paths at given distance	total number of associated bit errors
1	101	12	3	5	1	1
	111			6	2	4
2	10011	1,686	12	7	2	4
	11011			8	4	12
3	10011	1,817	9	7	3	7
	10111			8	3	10
4	10011	11,304	12	7	2	4
	11101			8	4	12

columns 5, 6, and 7 both error events at the free distance and error events at the free distance plus one are considered. In a binary comparison with the no-error sequence, an error event at distance $2k$ has the same probability of occurrence [4] as an error event at distance $2k - 1$, $k = 1, 2, \cdots$. Thus, in the case that the free distance is odd, error events at the free distance plus one should also be considered when comparing codes as to the bit error probability P_b. Studying columns 5, 6, and 7 of Table III, we observe that as far as the bit error probability is concerned code 4 is indistinguishable from codes 2 and 3. However, the syndrome decoder for code 4 requires 11 304 ROM-locations and code 4 is thus, from a complexity point of view, inferior to codes 2 and 3.

The number of metric combinations increases rapidly with the constraint length ν of the code. Hence, for larger values of ν the size of the ROM in the implementation according to Section IV soon becomes prohibitive. In the classical implementation [2] with a metric register and a path register for each state S_j, $j = 0, 1, \cdots, 2^\nu - 1$, the Viterbi decoder and the syndrome decoder require roughly the same amount of hardware per state. However, comparing (11b) and (11d) for code 1, one observes that S_1 and S_3 have the same metric value. Moreover, selecting the identical survivor in case of a tie, S_1 and S_3 also have the same path register contents. As far as metric and path register contents are concerned, the states S_1 and S_3 are not distinct. The metric register and the path register of either state S_1 or state S_3 can be eliminated. Thus, in the classical implementation with metric registers and path registers, the syndrome decoder for code 1 requires only $\frac{3}{4}$ of the amount of hardware that the Viterbi decoder requires. We will prove that, in general, one can eliminate the metric and path registers of half the odd numbered states, where the state number of a representative state $S = [0, s_2(\alpha)]$ is the value of the contents of the bottom register of the syndrome former interpreted as a binary number, i.e., odd states have $s_{2,-1} = 1$. Hence, the syndrome decoder is at most $\frac{3}{4}$ as complex as is the Viterbi decoder. Looking again at Table III column 4 we see that code 1 can be realized with 3 instead of $2^\nu = 4$ path registers, and that codes 2 and 4 can be realized with 12 instead of $2^\nu = 16$ path registers. Code 3 requires even fewer, i.e., nine instead of $2^\nu = 16$ path registers. We now prove that the syndrome decoder is, in general, at most $\frac{3}{4}$ as complex as is the Viterbi decoder.

Theorem 4: The odd numbered (representative) states $S(1) = [0,s_2(\alpha)]$ and $S'(1) = [0,s_2'(\alpha)]$, where $s_2'(\alpha) = s_2(\alpha) + s_2{}^b(\alpha) + \alpha^{-1}$ have identical metric equations, compare (11b) and (11d), iff $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$.

Proof: An odd numbered state $S(1) = [0,s_2(\alpha)]$ has $s_{2,-1} = 1$. If both connection polynomials $C_1(\alpha)$ and $C_2(\alpha)$ have a nonzero term of degree ν, it follows from the construction of the base state S^b that $s_{2,-1}{}^b = 1$. Hence, $s_2'(\alpha) = s_2(\alpha) + s_2{}^b(\alpha) + \alpha^{-1}$ has $s_{2,-1}{}' = 1$, and the requirement that both $S(1)$ and $S'(1)$ are odd numbered states is consistent. Consider the following parent states:

$$S_a(0) = [0,\alpha s_2(\alpha)]_{-\nu}{}^{-1}$$

$$S_b(0) = [0,\alpha^{-\nu} + \alpha s_2(\alpha)]_{-\nu}{}^{-1}$$

$$S_c(0) = [0,\alpha s_2'(\alpha)]_{-\nu}{}^{-1}$$

$$S_d(0) = [0,\alpha^{-\nu} + \alpha s_2'(\alpha)]_{-\nu}{}^{-1}.$$

As both $C_1(\alpha)$ and $C_2(\alpha)$ have a nonzero term of degree ν, $S_a(0)$ and $S_b(0)$ give rise to complementary syndrome digits, and so do $S_c(0)$ and $S_d(0)$. For an input $[n_{10},n_{20}] = [0,1]$ the parent states $S_a(0)$ and $S_b(0)$ go into $S(1)$ and the parent states $S_c(0)$ and $S_d(0)$ go into $S'(1)$, and vice versa for an input $[n_{10},n_{20}] = [1,0]$. Assuming that $C_1(\alpha)$ and $C_2(\alpha)$ both have a nonzero constant term and that $[n_{10},n_{20}]$ is either $[0,1]$ or $[1,0]$, the syndrome value only depends on the parent state $S(0)$. Hence, $S(1)$ and $S'(1)$ have identical equations. Q.E.D.

Theorem 4 proves that the syndrome decoders for the $\nu = 4$ codes 2, 3, and 4 in Table III can be realized with no more than 12 instead of $2^\nu = 16$ path registers. Column 4 of Table III shows, however, that code 3 requires only 9 instead of $2^\nu = 16$ path registers. The following theorem shows how this further reduction in hardware can be accomplished.

Theorem 5: One 4-tuple of pairs of odd numbered parent states gives rise to two pairs of odd numbered states that have identical metric equations iff $C_{1,1} = C_{2,1}$ and $C_{1,\nu-1} = C_{2,\nu-1}$.

Proof: Assume that $S_a(0)$, $S_b(0)$, $S_c(0)$, and $S_d(0)$ in the proof of Theorem 4 are odd numbered states, i.e., $s_{2,-2} = s_{2,-2}{}' = 1$. This can be a consistent requirement if $s_{2,-2}{}^b = 0$, and from the construction of the base state S^b it is clear that $s_{2,-2}{}^b = 0$ iff $C_{1,\nu-1} = C_{2,\nu-1}$. If $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ are odd numbered states, then according to Theorem 4 there exists a corresponding 4-tuple of odd numbered states $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ such that the corresponding components of these 4-tuples have identical metrics. According to Theorem 4, the first 4-tuple $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ gives rise to the states $S_p(1)$ and $S_q(1)$ that have identical metric equations. Similarly, according to Theorem 4 the second 4-tuple $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ gives rise to the states $S_p'(1)$ and $S_q'(1)$ that also have identical metric equations. As the corresponding members of the parent 4-tuples $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ and $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_{d'}(0)$ have identical metrics, the four states $S_p(1)$, $S_q(1)$, $S_p'(1)$, and $S_q'(1)$ have identical metric equations iff corresponding states in the parent 4-tuples give rise to the same syndrome digit. For this to occur it is necessary that the difference $S = [0,s_2{}^b(\alpha) + \alpha^{-1}]$ between corresponding states is a zero-equivalent state. But the base state $S^b = [\alpha^{-1},s_2{}^b(\alpha)]$ is a zero-equivalent state, hence, $S = [\alpha^{-1},s_2{}^b(\alpha)] + [\alpha^{-1},\alpha^{-1}]$ is a zero-equivalent state iff $C_{1,1} = C_{2,1}$. Q.E.D.

It is easy to verify that the two 4-tuples $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ and $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ in Theorem 5 with $[n_{10},n_{20}]$ equal to either $[0,0]$ or $[1,1]$ lead to four even numbered states that have pairswise-identical metric equations.

Corollary: The 4-tuple of pairs of odd numbered parent states of Theorem 5 gives rise to 4 even numbered states that have pairswise identical metric equations.

Summarizing, we have the following results. According to Theorem 4, the odd numbered states have pairswise-identical metric equations for codes, such as codes 1, 2, and 4 of Table III, with $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$. Hence, for such codes $2^{\nu-2}$, $\nu = 2, 3, \cdots$, metric and path register combinations can be eliminated. This leads to a syndrome decoder of $\frac{3}{4}$ of the hardware complexity of the Viterbi decoder. According to Theorem 5, each 4-tuple of pairs of odd numbered states leads to two pairs of odd numbered states that have identical metric equations for codes, such as code 3 of Table, III, with $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$ and $C_{1,1} = C_{2,1} \cdot C_{1,\nu-1} = C_{2,\nu-1}$. Hence, for such codes an additional $2^{\nu-4}$, $\nu = 4, 5, \cdots$, metric and path register combinations can be eliminated. According to the corollary to Theorem 5, the 4-tuple of pairs of odd numbered states mentioned above also leads to four even numbered states that have pairswise-identical metric equations. This leads to an additional saving of $2.2^{\nu-4}$ metric and path register combinations. The total savings for codes with $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$ and $C_{1,1} = C_{2,1} \cdot C_{1,\nu-1} = C_{2,\nu-1}$ is thus equal to $2^{\nu-2} + 3.2^{\nu-4}$, $\nu = 4, 5, \cdots$. The resulting syndrome decoder has 9/16 of the hardware complexity of the Viterbi decoder. Continuing this series of reductions the ultimate savings in metric and path register combinations is equal to $2^{\nu-2} + 3.2^{\nu-4} + 3^2.2^{\nu-6} + \cdots$ [9], leading to a syndrome decoder of hardware complexity equal to $(\frac{1}{2}\sqrt{3})^\nu$ times the hardware complexity of the Viterbi decoder. Note that in order to achieve this ultimate savings in hardware complexity one must put severe constraints on the encoder. Code 3 of Table III still achieves the maximum free distance for constraint length $\nu = 4$ codes. It is quite conceivable, however that in putting on further constraints on the encoder for larger values of ν it is no longer possible to achieve the maximum free distance. However, by only requiring $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$, and $C_{1,1} = C_{2,1} \cdot C_{1,\nu-1} = C_{2,\nu-1}$ one already achieves the reduction of 7/16, as shown above.

One final comment is in order. The hardware reduction with respect to the Viterbi decoder has been equated with the savings in metric and path register combinations. Note that the path registers in the snydrome decoder are used to store the binary ω-values, compare (5), i.e., they are binary storage registers just as are the path registers of the Viterbi decoder. One might remark that the path registers of the syndrome decoder are more complex because each state has four possible parent states instead of two as in the Viterbi decoder. However, by filling the path registers serially this aspect hardly adds to the complexity.

VI. EXPERIMENTAL RESULTS

The solid lines in Fig. 6 give the measured bit error probability P_b of code 3 of Table III as a function of the transition probability p of the binary symmetric channel (BSC), for both a path register length $D = 11$ and a path register length $D = 16$. The dashed line in Fig. 6 is a refinement of Van De Meeberg's [4] of Viterbi's upper bound on the bit error probability. This dashed bound is valid for infinite path register length. We extended Van De Meeberg's upper bound to also apply to finite path register lengths. The derivation of this extended bound will be published shortly. The dashed curves in Fig. 6 give the upper bound on the bit error probability for both $D = 11$ and $D = 16$. It is clear from Fig. 6 that it does not pay to increase the path register length much beyond $D = 16$.

It appears, so far, that the syndrome decoder is an interesting (from the hardware point of view) substitute for the classical Viterbi decoder. In closing, we want to mention two important applications of the syndrome decoder where the classical Viterbi decoder cannot be used. These applications are in feedback communications [5], and in source coding (data reduction) [6]. In the next two paragraphs we describe these uses, both of which have been simulated on the computer, of the syndrome decoder, respectively.

Reference [5] describes a coding strategy for duplex channels that enables one to transfer the hardware or the program complexity from the passive (receiving) side to the active (transmitting) side of the duplex channel. As pointed out in reference [5], this coding strategy can be used to great advantage in a computer network with a star configuration. For the information flow from the central computer to the satellites one uses the duplex strategy thus only requiring one complex one-way decoder at the central facility. For the information flow from a satellite computer towards the central facility one uses one-way coding, again using the complex one-way decoder at the central computer. One thus saves a number of complex one-way decoders equal to the number of satellite computers in the multiple dialog system (MDS). The duplex strategy [5] requires at the active (transmitting) side of the duplex channel an estimate of the forward noise $n_1(\alpha)$. To form this estimate $\hat{n}_1(\alpha)$, the data received at the passive station are scrambled by a convolutional scrambler $C(\alpha)$ and sent back to the active station. At the active station one can now form the estimate $\hat{n}_1(\alpha)$ using the Viterbi decoder for the "systematic" convolutional code generated by an encoder with connection polynomials $C_1(\alpha) = 1$, $C_2(\alpha) = C(\alpha)$. It is well known that "nonsystematic" convolutional codes are more powerful than systematic convolutional codes. With our syndrome decoder we are now able to find $\hat{n}_1(\alpha)$ according to a nonsystematic code. To this end we modify the feedback-free scrambler $C(\alpha)$ [5] into a feedback scrambler $C_2(\alpha)/C_1(\alpha)$, see Fig. 7. Note that $z(\alpha)$ according to Fig. 7 is identical to (1). Hence, we can use our syndrome decoder to obtain $\hat{n}_1(\alpha)$. Fig. 8 gives the feedback scrambler for the convolutional code generated by the encoder of Fig. 1.

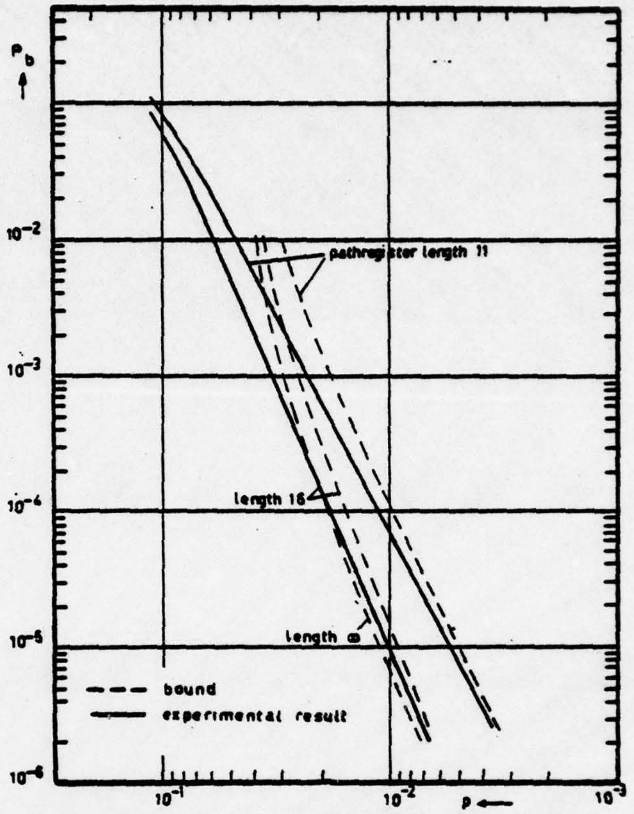


Fig. 6. Bit error rate P_b versus channel transition probability p.

Note that the syndrome former, Fig. 2, has two input sequences $n_1(\alpha)$, $n_2(\alpha)$ and one output sequence $z(\alpha) = C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha)$. Thus, the syndrome former compresses two binary streams $n_1(\alpha)$, $n_2(\alpha)$ into one stream $z(\alpha)$ and, hence, achieves a data compression of a factor of 2. The estimator part of the syndrome decoder can with high probability of being correct recover the original sequences $n_1(\alpha)$, $n_2(\alpha)$ given the compressed data sequence $z(\alpha)$. The use of a syndrome decoder for data compression has also been studied by Massey [6]. In general, to obtain a data compression factor n, $n = 2, 3, \cdots$, one used the syndrome decoder of a rate $-(n - 1)/n$ convolutional code.

VII. CONCLUSIONS

This paper considers the syndrome decoding of rate-$\frac{1}{2}$ convolutional codes. Table III shows that the number of metric combinations of the syndrome decoder is small compared to the number of metric combinations of the corresponding Viterbi decoder. For the constraint length $\nu = 4$ code of row 3 of Table III, for example, the number of metric combinations with syndrome decoding is 1817, whereas the Viterbi decoder for this same code has over 15 000 metric

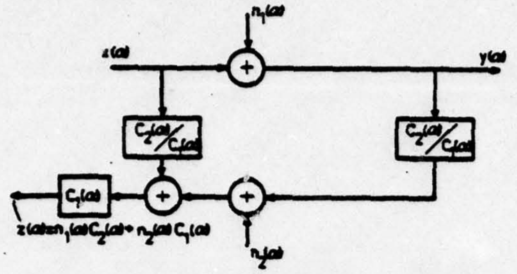

$n_1(\alpha)$
$x(\alpha)$
$y(\alpha)$
$C_2(\alpha)/C_1(\alpha)$
$C_2(\alpha)/C_1(\alpha)$
$C_1(\alpha)$
$z(\alpha)=n_1(\alpha)C_2(\alpha)+n_2(\alpha)C_1(\alpha)$
$n_2(\alpha)$

Fig. 7. Duplex channel with feedback.scramblers.

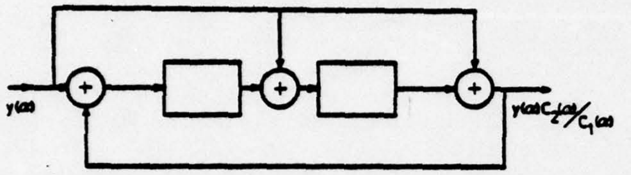


Fig. 8. Feedback scrambler with $C_1(\alpha) = 1 + \alpha^2$, $C_2(\alpha) = 1 + \alpha + \alpha^2$.

combinations. This relatively small number of metric combinations for small constraint length codes enables the ROM-implementation of Section IV, that eliminates the need for metric registers. For larger constraint lengths, the storage requirements of the ROM would become excessive. However, by putting mild constraints on the encoder it is possible to eliminate more than half of the metric and path register combinations. The syndrome decoder of code 3 of Table III, for example, only requires nine path registers, whereas the corresponding Viterbi decoder has $2^\nu = 16$ path registers.

The idea of syndrome decoding can be extended to rate $-k/n$ convolutional codes. Forney [7], [8] describes the mathematical tools necessary to find the general syndrome former equations and the equations of the inverse encoder.

Note added in proof: A. W. J. Kolen has pointed out a mistake in the proof of Theorem 3, i.e., $D_1'(\alpha)C_1(\alpha) + D_2'(\alpha)C_2(\alpha) = 0$ is incorrect. The proof can be corrected by observing that deg $[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)] >$ deg $[D_1(\alpha)s_1(\alpha) + D_2(\alpha)s_2(\alpha)]$.

ACKNOWLEDGMENT

The authors want to thank L. J. A. E. Rust for his encouragement and help with the computer simulations.

REFERENCES

[1] J. P. M. Schalkwijk and A. J. Vinck, "Syndrome decoding of convolutional codes," *IEEE Trans. Commun.* (Corresp.), vol. COM-23, pp. 789–792, July 1975.

[2] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol. (Special Issue on Error Correcting Codes—Part II)*, vol. COM-19, pp. 751–772, Oct. 1971.

[3] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[4] L. Van De Meeberg, "A tightened upper bound on the error probability of binary convolutional codes with Viterbi decoding," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 389–391, May 1974.

[5] J. P. M. Schalkwijk, "A coding scheme for duplex channels," *IEEE Trans. Commun. (Special Issue on Communications in Europe)*, vol. COM-22, pp. 1369–1374, Sept. 1974.

[6] J. L. Massey, "The codeword and syndrome methods for data compression with error-correcting codes," in *Proc. NATO Advanced Study Institute on New Directions in Signal Processing in Communications and Control*, Darlington, England, Aug. 5–17, 1974.

[7] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, November 1970; also, correction appears in vol. IT-17, p. 360, May 1971.

[8] —, "Structural analysis of convolutional codes via dual codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 512–518, July 1973.

[9] J. P. M. Schalkwijk, "Symmetries of the state diagram of the syndrome former of a binary rate-½ convolutional code," Lecture Notes, CISM Udine Summer School on Coding, Udine, Italy, Sept. 2–12, 1975.

★

J. Pieter M. Schalkwijk (M'66) was born in Rijswijk, The Netherlands, on November 1, 1936. He received the M.S. degree in electrical engineering from the Technological University, Delft, The Netherlands, in 1959, and the Ph.D. degree, also in electrical engineering, from Stanford University, Stanford, CA, in 1965.

From 1959 to 1961, while in military service in The Netherlands, he was assigned to work at Philips, Hengelo, The Netherlands, on the control problems that arise when using a digital computer as part of a radar fire-control system. In 1961 he joined the National Defense Research Laboratory of The Netherlands, where he carried out comparative tests of HF digital-data terminals. In 1963 he worked on windtunnel measurements at the National Aeronautics and Space Research Laboratories in Amsterdam, The Netherlands. In 1963 he came to the U.S. to continue his advanced study. In 1965 he joined the Applied Research Laboratory of Sylvania Electronic Systems, Waltham, MA, where he worked on problems of signal design and reception for digital-data transmission. From 1968 to 1972 he was an Assistant Professor of Information and Computer Science at the University of California at San Diego. Since 1972 he has been a Professor of Communication Theory at the Technological University, Eindhoven, The Netherlands.

Dr. Schalkwijk is a member of The Netherlands Electronics and Radio Society, The Royal Netherlands Institute of Engineers, and the Association for Computing Machinery.

★

A. J. Vinck was born in Breda, The Netherlands, on May 15, 1949. He received the M.S. degree in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1974.

He is currently working at the Department of Electrical Engineering, Eindhoven University of Technology, where he is involved in research on convolutional codes.

END

DTIC

6-86